CMPUT463/563
Probabilistic Graphical Models

Approximate Inference: Sampling

Lili Mou

Dept. Computing Science, University of Alberta

lmou@ualberta.ca

Outline

- MC Sampling: CDF sampling, rejection sampling, importance sampling, self-normalized IS
- MCMC: dependent sampling
  - Gibbs sampling (posterior sampling)
  - Metropolis—Hastings
- Applications
  - Restricted Boltzmann machine
  - Unsupervised text generation (not required for exams)

Sampling

- Sampling is widely used to estimate certain quantities that involve randomness
- E.g.: estimate P(head) of a coin
  - It's arguable the classical mechanics is deterministic, as no quantum effect is expected in this process.
  - The outcome of tossing a coin is a deterministic function of initial velocity, position, angle, gravity, etc. Thus, P(head) is a function of P(init velocity), P(init pos), P(init angle), etc. However, computing such a function is intractable.
- A much easier approach is to toss the coin M times, with outcomes $X^{(1)}, X^{(2)}, \cdots, X^{(M)}$. Then,

$$\widehat{\Pr}(head) = \frac{1}{M} \sum_{m=1}^{M} X^{(m)}$$

Forward (Causal) Sampling

- Sample a variable conditioned on its parent(s)
  - Topologically sort all variables
  - Sample $X_i \sim P(X_i \,|\, \mathrm{Par}(X_i))$
- Works well for Bayesian networks in the unconstrained case
- Does not work for MN and constrained BN in general
- We need more efforts in investigating sampling methods

Monte Carlo: Sampling from CDF

- Probabilistic density function (PDF) $\Pr[a \leq x \leq b] = \int_a^b f(x)\, dx$
- Cumulative density function (CDF) $F(x) = \int_{-\infty}^{x} f(u)\, du = \Pr[u \leq x]$
- Sampling procedure $u \sim \mathrm{U}[0,1]; \quad x = \mathrm{CDF}^{-1}(u)$

Main drawbacks

- CDF not analytic, especially the conditional CDF for multivariate distributions
- **Fun facts:** In computer science, randomized algorithms can be divided to
  - **Las Vegas** algorithms: always give the correct result. E.g., randomized quick sort
  - **Monte Carlo** algorithms: may (usually) generate wrong results. E.g., in machine learning, a Monte Carlo approach usually refers to sampling methods

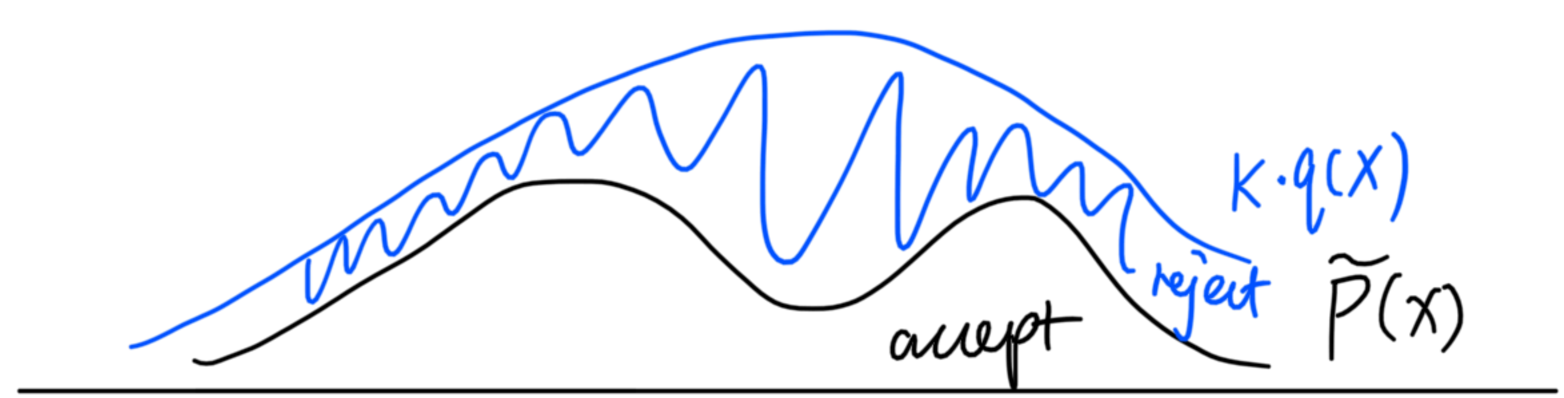
MC: Rejection Sampling

- To sample from $p(x) = \frac{1}{Z}\widetilde{p}(x)$
- We instead sample $x \sim q(x)$
- Accept the sample x with probability $\dfrac{\widetilde{p}(x)}{k \cdot q(x)}$

  where k is a constant s.t. $kq(x) \geq \widetilde{p}(x), \forall x$

- Reject x w.p. $1 - \dfrac{\widetilde{p}(x)}{k \cdot q(x)}$

Efficiency?



MC: Importance Sampling

- Goal: To compute $\mathbb{E}_{x \sim p(x)}[f(x)]$
- Instead of sampling $x \sim p(x)$, we plan to sample from $q(x)$

$$\mathbb{E}_{x \sim p(x)}[f(x)] = \int f(x) p(x) dx$$

$$= \int f(x) \frac{p(x)}{q(x)} q(x) dx$$

$$= \mathbb{E}_{x \sim q(x)}[f(x) w(x)]$$

where $w(x) = \dfrac{p(x)}{q(x)}$ in the importance weight

$$\approx \frac{1}{M} \sum_{m=1}^{M} f(x^{(m)}) w(x^{(m)})$$

MC: Self-Normalized IS

- Goal: To compute $\mathbb{E}_{x \sim p(x)}[f(x)]$ with only $\tilde{p}(x)$

- We still sample from $q(x)$

$$\mathbb{E}_{x \sim p(x)}[f(x)] \quad = \frac{\mathbb{E}_{x \sim q}[f(x)w(x)]}{\mathbb{E}_{x \sim q}[w(x)]}$$

where importance weight $w(x) = \dfrac{\tilde{p}(x)}{\tilde{q}(x)}$

may be given by unnormalized measures

HW: Proof the above equation and give a Monte Carlo estimate

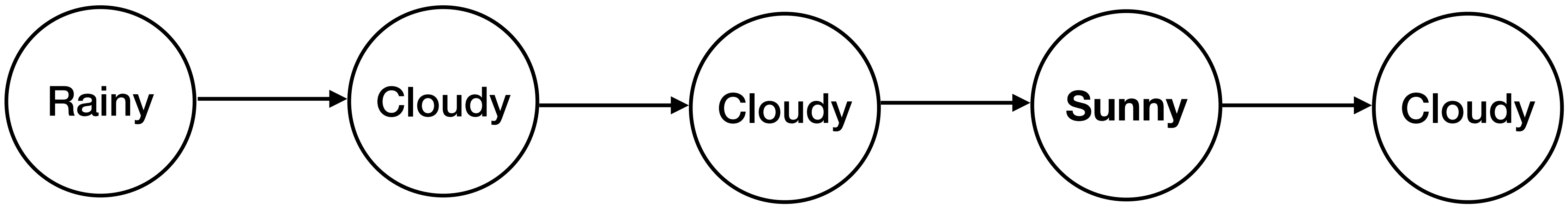
Markov Chain Monte Carlo

- Monte Carlo (MC) sampling is also known as **independent** sampling. If $X^{(1)}, X^{(2)}, \cdots, X^{(M)}$ are MC samples, then they are iid.

- MC sampling may still be inadequate, e.g., PGM inference questions. Thus, Markov Chain Monte Carlo (MCMC) methods are developed.

- MCMC is also known as **dependent** sampling. If $X^{(1)}, X^{(2)}, \cdots, X^{(M)}$ are MCMC samples, then
  - They are dependent sampling
  - Ideally, $X^{(M)}$ shall converge to an unbiased sample from a desired distribution

Markov Chain



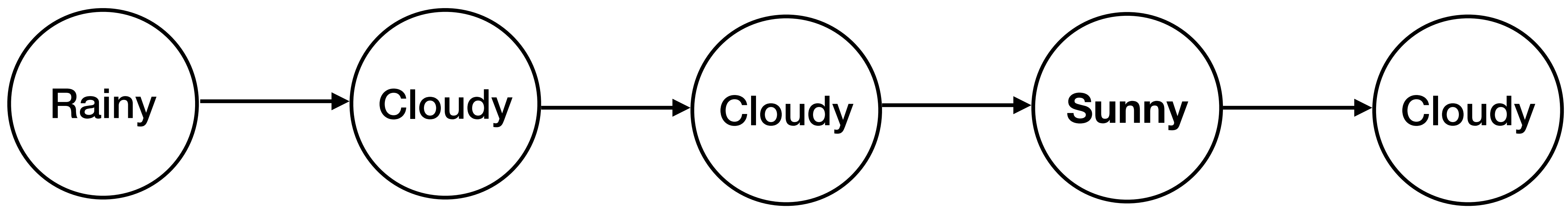
- Markov Chain
  - At step t, the state probability is $\boldsymbol{p}^{(t)}$, i.e.,

    $$p_i^{(t)} = \Pr[s^{(t)} = i]$$

  - Transition probability T, i.e., $T_{ij} = \Pr[s^{(t+1)} = j \mid s^{(t)} = i]$

- Observations
  - T is normalized in each row
  - The probability of the next step $\boldsymbol{p}^{(t+1)} = T^\top \boldsymbol{p}^{(t)}$

A Very Difficult Theorem



- **Thm.** Regular Markov chains converges to a unique stationary distribution. (Not proved in this course)
  - Regular Markov chain: There exists some k such that for every two states s, s', the probability of going from s to s' with exactly k steps is non-zero.
- Applying MC to sampling:
  - Design a MC such that the stationary distribution is the desired distribution to sample from. Sampling follows the MC.

Gibbs Sampling

- Gibbs sampling (sampling from the posterior)
  - Pick an arbitrary $\boldsymbol{X}^{(0)} = (X_1^{(0)}, \cdots, X_n^{(0)})$
  - For $t = 1,2,\cdots,$
    - Pick $i \in \{1,2,\cdots,n\}$
    - Sample $X_i^{(t)} \sim P(X_i \mid \boldsymbol{X}_{-i}^{(t-1)})$
    - $\boldsymbol{X}^{(t)} = (X_1^{(t-1)}, \cdots, X_{i-1}^{(t-1)}, X_i^{(t)}, X_{i+1}^{(t-1)}, \cdots, X_n^{(t-1)})$

$\boldsymbol{X}_{-i}$ refers to all variables except X_i

Gibbs Sampling

- Computing posterior is efficient in PGMs
  - The posterior of a variable only concerns its neighbors

$$P(X_i \mid \boldsymbol{X}_{-i}) = \frac{\prod_{\phi:i\in \mathrm{scope}(\phi)} \phi(\boldsymbol{X}) \cdot \cancel{\prod_{\phi:i\notin \mathrm{scope}(\phi)} \phi(\boldsymbol{X}_{-i})}}{\sum_{X_i'} \prod_{\phi:i\in \mathrm{scope}(\phi)} \phi(\boldsymbol{X}') \cdot \cancel{\prod_{\phi:i\notin \mathrm{scope}(\phi)} \phi(\boldsymbol{X}_{-i})}}$$

- Correctness. To show the joint distribution $P(X_i, \boldsymbol{X}_{-i})$ is satisfies the stationary property:

$$P(\boldsymbol{X}') = \sum_{X} P(X)P(\boldsymbol{X}' \mid \boldsymbol{X})$$

Consider any $X' = (X_1', \cdots, X_n')$. Then, X can only differ by X_i. We assume $X = (X_1', \cdots, X_i, \cdots X_n')$.

$$\mathrm{RHS} = \sum_{X_i} \underbrace{P(X_i, \boldsymbol{X}_{-i}')}_{= P(X)} P(X_i' \mid \boldsymbol{X}_{-i}') = P(\boldsymbol{X}_{-i}')P(X_i' \mid \boldsymbol{X}_{-i}') = \mathrm{LHS}$$

Metropolis—Hastings Sampler

- **Input**
  - An arbitrary desired distribution $p(x)$
- **Output**
  - An unbiased sample $x \sim p(x)$
- **Algorithm**
  - Start from an arbitrary initial state $x^{(0)}$
  - For every step t — $g(x' \mid x)$: arbitrary proposal distribution

    Propose a new state $x' \sim g(x' \mid x^{(t)})$

    Accept x' w.p. $A(x' \mid x) = \min \left\{ 1, \dfrac{p(x')g(x^{(t)} \mid x')}{p(x)g(x' \mid x^{(t)})} \right\}$, i.e.,

    $x^{(t+1)} = x'$

    Reject x' otherwise, i.e., $x^{(t+1)} = x^{(t)}$
  - Return $x^{(t)}$ with a large t

Proof Sketch

- Detailed balance property = > Stationary distribution

If

$$\forall x, y, \qquad \pi(x) \cdot \mathcal{T}_{x \to y} = \pi(y) \cdot \mathcal{T}_{y \to x}$$

Then

$\pi(x)$ is a stationary distribution

Because

$$\forall x, \qquad \pi(x) = \sum_y \pi(y) \mathcal{T}_{y \to x} = \sum_y \pi(x) \mathcal{T}_{x \to y} = \pi(x)$$

Proof Sketch (Cont.)

MH Sampler satisfies detailed balance

- $\forall x, y,$ if $x \neq y,$ $$p(x) \cdot \mathcal{T}_{x \to y} = p(x) \cdot g(y|x) \cdot \min\left\{1, \frac{p(y)g(x|y)}{p(x)g(y|x)}\right\} \quad (1)$$
$$p(y) \cdot \mathcal{T}_{y \to x} = p(y) \cdot g(x|y) \cdot \min\left\{1, \frac{p(x)g(y|x)}{p(y)g(x|y)}\right\} \quad (2)$$

- W.L.O.G., we assume $p(x)g(y|x) \geq p(y)g(x|y)$

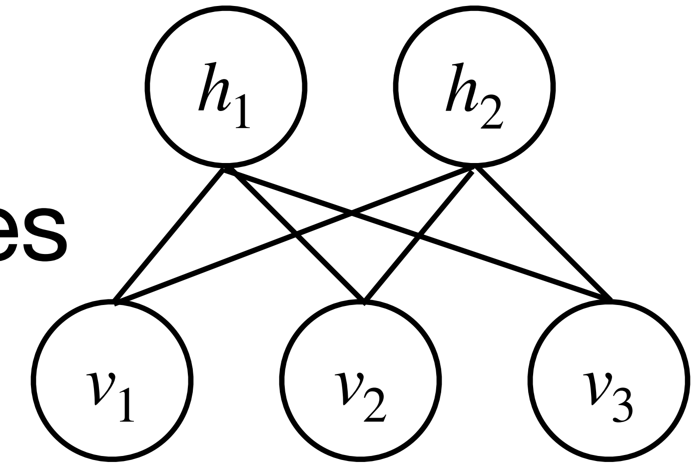
$$(1) = p(y) \cdot g(x|y)$$

$$(2) = p(y) \cdot g(x|y)$$

- $\forall x, y,$ if $x = y,$ $p(x)\mathcal{T}_{x \to y} = p(y)\mathcal{T}_{y \to x}$ also holds

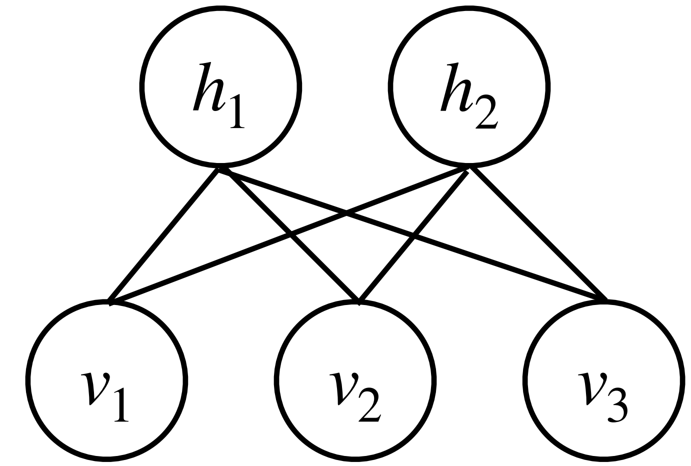
Applications: Restricted Boltzmann Machine

Restricted Boltzmann Machine



- $\boldsymbol{v}$: observable variables; $\boldsymbol{h}$: hidden variables
- EM training:
  - Gradient $\mathbb{E}_{\boldsymbol{h} \sim P(\boldsymbol{h}|\boldsymbol{v})}[f_i] - \mathbb{E}_{\boldsymbol{h},\boldsymbol{v} \sim P(\boldsymbol{h},\boldsymbol{v})}[f_i]$
  - First term is easy
  - Second term can be estimated by MCMC.
    - In principle, Gibbs sampling should work in sequence. However, since $v_i \perp v_j | \boldsymbol{h}$, we can sample $\boldsymbol{v} | \boldsymbol{h}$ in parallel; since $h_i \perp h_j | \boldsymbol{v}$, we can sample $\boldsymbol{h} | \boldsymbol{v}$ in parallel

Applications: Restricted Boltzmann Machine



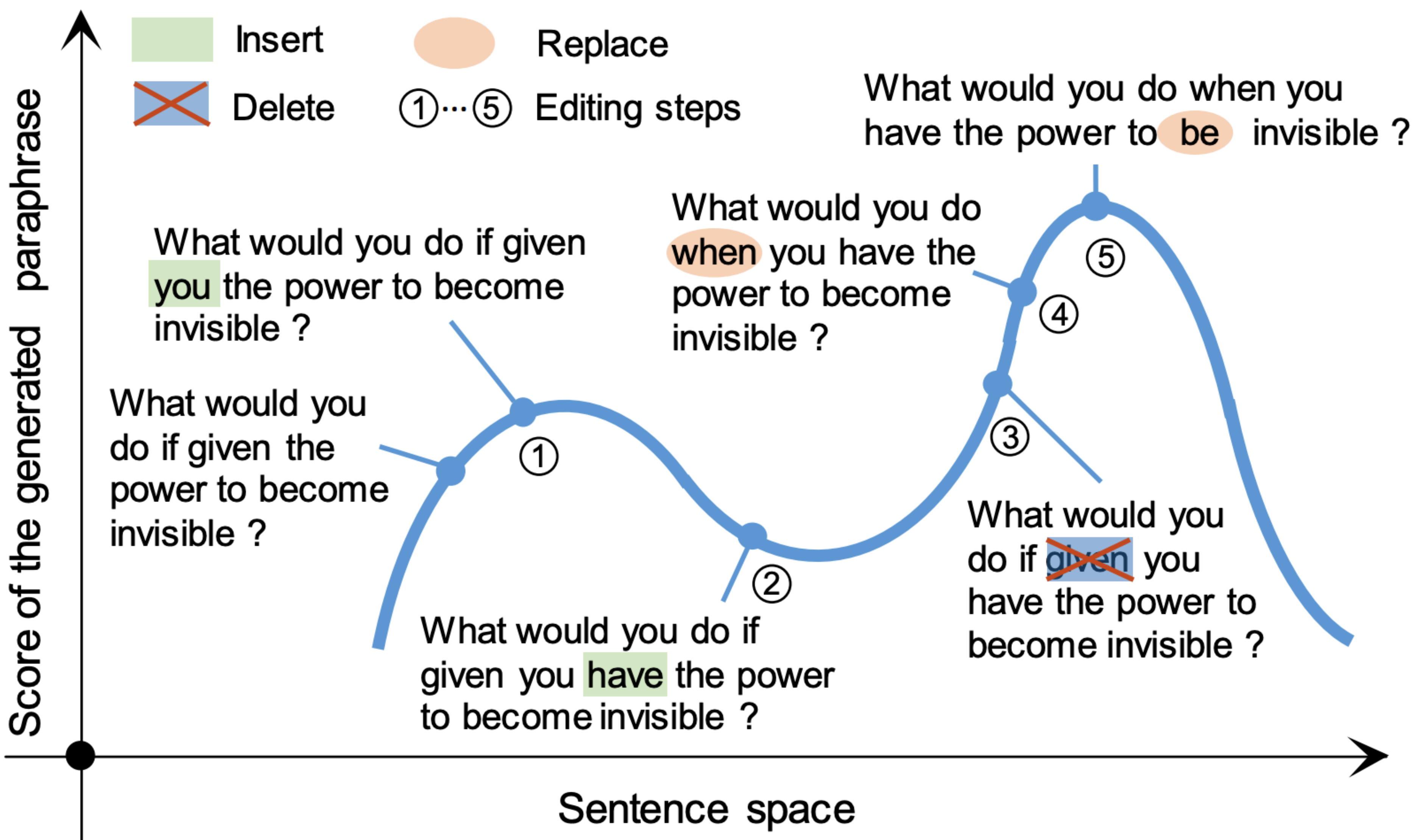
- Given observed $\boldsymbol{v}^{(0)}$, we should perform
  - $\boldsymbol{h}^{(1)} \sim P(\boldsymbol{h} \,|\, \boldsymbol{v}^{(0)})$
  - $\boldsymbol{v}^{(1)} \sim P(\boldsymbol{v} \,|\, \boldsymbol{h}^{(1)})$
  - …
  - Until $\boldsymbol{h}^{(\infty)}, \boldsymbol{v}^{(\infty)}$ for estimating $\mathbb{E}_{\boldsymbol{h},\boldsymbol{v} \sim P(\boldsymbol{h},\boldsymbol{v})}[f_i]$
- Hinton's contrastive divergence (CD)
  - CD-n uses $\boldsymbol{h}^{(n)}, \boldsymbol{v}^{(n)}$ to estimate $\mathbb{E}_{\boldsymbol{h},\boldsymbol{v} \sim P(\boldsymbol{h},\boldsymbol{v})}[f_i]$
  - CD-1 uses $\boldsymbol{h}^{(1)}, \boldsymbol{v}^{(1)}$

Applications: Unsupervised Text Generation

- **Search objective**
  - Scoring function measuring text quality

$$s(\mathbf{y}) = s_{LM}(\mathbf{y}) \cdot s_{Semantic}(\mathbf{y})^{\alpha} \cdot s_{Task}(\mathbf{y})^{\beta}$$

- **Search algorithm**
  - Currently we are using stochastic local search



Search Algorithm

Local edits for $\mathbf{y}' \sim \text{Neighbor}(\mathbf{y}_t)$

- General edits
  - Word deletion
  - Word insertion
  - Word replacement

$$p(w_*|\cdot) = \frac{f_{\text{sim}}(\mathrm{x}_*, \mathrm{x}_0) \cdot f_{\text{exp}}(\mathrm{x}_*, \mathrm{x}_0) \cdot f_{\text{flu}}(\mathrm{x}_*)}{Z},$$
$$Z = \sum_{w_* \in \mathcal{W}} f_{\text{sim}}(\mathrm{x}_*, \mathrm{x}_0) \cdot f_{\text{exp}}(\mathrm{x}_*, \mathrm{x}_0) \cdot f_{\text{flu}}(\mathrm{x}_*),$$

Gibbs in Metropolis

- Task specific edits
  - Reordering, swap of word selection, etc.

Search Algorithm

Example: Metropolis—Hastings sampling

Start with $\mathbf{y}_0$ # an initial candidate sentence

Loop within budget at step t:

$\mathbf{y}' \sim \text{Neighbor}(\mathbf{y}_t)$ # a new candidate in the neighbor

Either reject or accept $\mathbf{y}'$

$$\boxed{\begin{aligned} A(\mathrm{x}'|\mathrm{x}_{t-1}) &= \min\{1, A^*(\mathrm{x}'|\mathrm{x}_{t-1})\} \\ A^*(\mathrm{x}'|\mathrm{x}_{t-1}) &= \frac{\pi(\mathrm{x}')g(\mathrm{x}_{\mathrm{t}-1}|\mathrm{x}')}{\pi(\mathrm{x}_{t-1})g(\mathrm{x}'|\mathrm{x}_{t-1})} \end{aligned}}$$

If accepted, $\mathbf{y}_t = \mathbf{y}'$, or otherwise $\mathbf{y}_t = \mathbf{y}_{t-1}$

Return the best scored $\mathbf{y}_*$

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, Lei Li. CGMH: Constrained sentence generation by Metropolis-Hastings sampling. In AAAI, pages 6834--6942, 2019.

Search Algorithm

Example: Simulated annealing

Start with $\mathbf{y}_0$ # an initial candidate sentence

Loop within budget at step t:

$\mathbf{y}' \sim \text{Neighbor}(\mathbf{y}_t)$ # a new candidate in the neighbor

Either reject or accept $\mathbf{y}'$ $\boxed{p(\text{accept}|\mathrm{x}_*, \mathrm{x}_t, T) = \min\left(1, e^{\frac{f(\mathrm{x}_*) - f(\mathrm{x}_t)}{T}}\right)}$

If accepted, $\mathbf{y}_t = \mathbf{y}'$, or otherwise $\mathbf{y}_t = \mathbf{y}_{t-1}$

Return the best scored $\mathbf{y}_*$

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, Sen Song. Unsupervised paraphrasing by simulated annealing. In ACL, pages 302--312, 2020.