Probabilistic Graphical Models (4): sampling-based approximate inference

Qinfeng (Javen) Shi

The Australian Centre for Visual Technologies,
The University of Adelaide, Australia

10 June 2011

Course Outline

Probabilistic Graphical Models:

1. Representation
2. Inference
3. Learning
4. Sampling-based approximate inference (Today)
5. Temporal models
6. . . .

Sampling Outline

- Understanding samples
- Sampling techniques overview
- Sampling techniques in PGM inference

Understanding samples

In fact, there is no way to check 'a sample' is from a distribution or not — two totally different distributions can generate the same sample. For example, $uniform[0,1]$ and gaussian $N(0,1)$ can both generate a sample with value 0. Looking at a sample with value = 0 alone, how do you know its distribution for sure? What we really check (and know for sure) is the way that the samples were generated. When we say a procedure generates a sample from a distribution P, what we really mean is that keeping sampling this way (by the procedure), the normalised histogram H^n with n samples is going to converge to the distribution P. That is $H^n \to P$ as $n \to \infty$. If we don't know the way that the samples were generated, we never know what's the distribution for sure — we can only guess (e.g. using statistical tests) based on a number of available samples.

Sampling techniques overview

- Monte Carlo
- Importance sampling
- Acceptance-rejection sampling
- Markov chain Monte Carlo (MCMC)

Monte Carlo

Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to compute their results.

```
repeat
    draw sample(s)
    compute result according to the samples
until sampled enough ( or the result is stable)
```

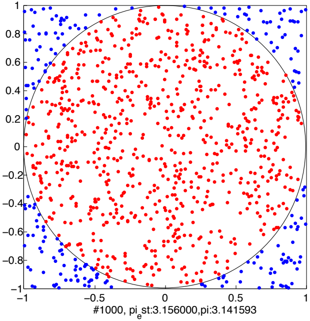
Monte Carlo

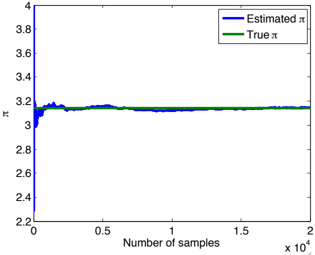
To estimate π (area of a circle with radius r is $S_c = \pi r^2$).
Idea:

- draw a circle ($r = 1$) and a rectangle ($2r \times 2r$) enclosing the circle. We know the area of the rectangle is $S_{rec} = (2r)^2$. If we can estimate the area of the circle, then we can estimate π by $\pi = S_c/r^2$.
- Draw a sample point from the rectangle area uniformly. The chance of it being within the circle is S_c/S_{rec}. So if we throw enough points, we have $N_{within}/N_{total} \approx S_c/S_{rec}$. Thus $S_c \approx S_{rec}N_{within}/N_{total}$

See a matlab demo.

Monte Carlo


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1
−0.5
0
0.5
1
#1000, pi_est:3.156000,pi:3.141593


Estimated π
True π
4
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
π
0
0.5
1
1.5
2
Number of samples
x 10^4

Monte Carlo

To estimate an expectation:
Generate samples $x_i \sim q(X), i = 1, \ldots, N$.

$$\begin{aligned} \mathbb{E}_{X \sim q(X)}[f(X)] &\approx \hat{\mathbb{E}}_{X \sim q(X)}[f(X)] \\ &= \frac{1}{N} \sum_{i=1}^{N} f(x_i), \end{aligned}$$

Importance sampling

To compute $\mathbb{E}_{X \sim p(X)}[f(X)]$.
Assume $p(x)$ (target distribution) is hard to sample from directly, and $q(x)$ (proposal distribution) is easy to sample from and $q(x) > 0$ when $p(x) > 0$.

$$\begin{aligned}\mathbb{E}_{X \sim p(X)}[f(X)] &= \int_x p(x) f(x) dx \\ &= \int_x q(x) \frac{p(x)}{q(x)} f(x) dx \\ &= \mathbb{E}_{X \sim q(X)}[\frac{p(X)}{q(X)} f(X)].\end{aligned}$$

$$\hat{\mathbb{E}}_{X \sim p(X)}[f(X)] = \hat{\mathbb{E}}_{X \sim q(X)}[\frac{p(X)}{q(X)} f(X)],$$

where $\hat{\mathbb{E}}_{X \sim q(X)}[f(X)] = \frac{1}{N} \sum_{i=1}^{N} f(x_i), x_i \sim q(X), i = 1, \dots, N.$

Acceptance-rejection sampling

Target: to sample X from $p(x)$.
Given: $q(x)$ easy to sample from.
Find a constant M such that $M \cdot q(x) \geq p(x), \forall\ x$.

```
repeat
    step 1: sample $Y \sim q(y)$
    step 2: sample $U \sim Uniform[0, 1]$
    if $U \leq \frac{p(y)}{M \cdot q(y)}$ then
        then $X = Y$;
    else
        reject and go to step 1.
    end if
until sampled enough
```

Acceptance-rejection sampling

Proof:

$$\because Pr(accept|X = x) = \frac{p(x)}{M \cdot q(x)} \quad \text{and} \quad Pr(X = x) = q(x)$$

$$\therefore Pr(accept) = \int_x Pr(accept|X = x) \cdot Pr(X = x)dx$$

$$= \int_x \frac{p(x)}{M \cdot q(x)} \cdot q(x)dx = \frac{1}{M} \quad (\text{ thus don't want } M \text{ big})$$

$$\therefore Pr(X|accept) = \frac{Pr(accept|X) \cdot P(X)}{Pr(accept)}$$

$$= \frac{\frac{p(x)}{M \cdot q(x)} \cdot q(x)}{\frac{1}{M}} = p(x).$$

Understanding AR sampling (1)

I guess the most confusing part, is why M comes in. So let's look at the case without M first.
Denote the histogram formed by n samples from $q(x)$ as H_q^n, the histogram formed by n samples from $p(x)$ as H_p^n, the histogram formed by n accepted samples from AR sampling procedure as H^n.
For a sample $x \sim q(x)$, if $p(x) < q(x)$, it suggests if you accept all the x and keep sampling this way, the histogram you will get is H_q^n. But what you really want to get, is a way that the resulting histogram H becomes H_p^n. Rejecting some portion of x can make the histogram H has the same shape as H_p at point x. In other words, the histogram H has more counts at point x than H_p, so we remove some counts to make $H(x) = H_p(x)$. (Take a moment to think this through).

Understanding AR sampling (2)

What if for a sample $x \sim q(x), p(x) > q(x)$? The histogram H_q^n already has less counts than H_p^n at x. What do we do? Well, we can sample $M \times n$ points from $q(x)$ to build H_q^{Mn} first. Now H_q^{Mn} should have more counts than H_p^n at x (because we choose a M such that $p(x) < Mq(x)$ for all x. If not, choose a larger M). Visually, H_q^{Mn} encloses H_p^n. At point x, we only want to keep $H_p^n(x)$ many samples from totally $H_q^{Mn}(x)$ many. This is how uniform sampling and M came in. We sample $u \sim Uniform[0, Mq(x)]$, accept x when $u < p(x)$ (equivalent to sample $u \sim Uniform[0, 1]$, accept x when $u < p(x)/Mq(x)$). As a result, after Mn samples, we will get a H close to H_p^n. Moreover,

$$\lim_{n\to\infty} H^n = \lim_{n\to\infty} H_p^n = p.$$

Understanding AR sampling (3)

Here we can choose any M such that $p(x) < Mq(x)$ for all x. The bigger M is, the more samples (Mn samples) you need to approximate H_p^n. That's why in practice, people want to use the smallest M (such that $p(x) < Mq(x)$ for all x) to reduce the number of rejected samples.

Markov chain Monte Carlo

Sampling from probability distributions based on constructing a Markov chain that has the desired distribution $p(x)$ as its equilibrium distribution $\pi(x)$.

- Metropolis-Hastings algorithm
- Gibbs sampling
- …

Metropolis-Hastings algorithm

Ingredients:

- want to sample from $\pi(x)$ (but impossible directly).
- sample from $q(x)$ is easy.
- a homogenous and stationary Markov chain with transition kernel $q(x_{t+1}|x_t)$.

Metropolis-Hastings algorithm

Properties of Markov chain: let $(X_n)_{n\geq 0}$ be regular Markov (λ, P), then for all $n, m \geq 0$,

- $\Pr(X_n = j) = (\lambda P^{(n)})_j$
- exists an unique invariant (stationary) π', for any λ,

$$\Pr(X_n = j) \to \pi'_j \quad \text{as} \;\; n \to \infty \quad \text{for all} \;\; j$$

- If detailed balance equation holds,

$$\pi_i P_{ij} = \pi_j P_{ji},$$

π is the invariant distribution.

Metropolis-Hastings algorithm

We know that for a regular markov chain, given transition kernel q and initial distribution λ, sampling from the chain will eventually become sampling from its invariant distribution π'.
Metropolis-Hastings algorithm asks a reversed qestion: How do we change q, such that the invariant distribution becomes the desirable π instead of π'? That is, without knowing π', but knowing λ, q, we know there exists a π', such that $(\lambda q^{(n)}) \to \pi'$ as $n \to \infty$. Now, knowing π, λ, q, how do we find q' such that $(\lambda q'^{(n)}) \to \pi$ as $n \to \infty$?

Metropolis-Hastings algorithm

Suppose have x_t from $\pi(x)$, to sample x_{t+1} from $\pi(x)$.
Sample $x' \sim q(x|x_t)$ first.
Case 1: If $\pi(x_t)q(x'|x_t) = \pi(x')q(x_t|x')$ (detailed balance), take $x_{t+1} = x'$.
Case 2: if $\pi(x_t)q(x'|x_t) > \pi(x')q(x_t|x')$, it means x' too often. Need to accept it with probability α, such that $\pi(x_t)[\alpha q(x'|x_t)] = \pi(x')q(x_t|x')$. So $\alpha = \frac{\pi(x')q(x_t|x')}{\pi(x_t)q(x'|x_t)}$. Accept x' with probability α is simple: draw $u \sim \mathit{Uniform}[0,1]$. If $u < \alpha$, $x_{t+1} = x'$, else $x_{t+1} = x_t$ (so can resample $x' \sim q(x|x_t)$ again).
Case 3: if $\pi(x_t)q(x'|x_t) < \pi(x')q(x_t|x')$, it means x' too few. So accept all x'. That is $x_{t+1} = x'$.

Metropolis-Hastings algorithm

```
Target: to sample from $\pi(x)$.
for $t = 1, 2, \cdots, N$ do
    Generate $x' \sim q(x|x_t)$, $u \sim Uniform[0, 1]$
    $A(x_t \to x') = \min\{1, \frac{\pi(x')q(x_t|x')}{\pi(x_t)q(x'|x_t)}\}$
    if $u \le A(x_t \to x')$ then
        $x_{t+1} = x'$;
    else
        $x_{t+1} = x_t$;
    end if
end for
```

So $q'(x'|x_t) = A(x'|x_t)q(x'|x_t)$. One can check for any q

$$\pi(x_t)q'(x'|x_t) = \pi(x')q'(x_t|x').$$

So one can build $Markov(\lambda, q')$ from any q (as long as q makes it regular), any λ , such that $(\lambda q'^{(n)}) \to \pi$.

Metropolis-Hastings algorithm

Note a regular *Markov*(λ, q') only essures $(\lambda q'^{(n)}) \rightarrow \pi$ as $n \rightarrow \infty$. Thus we need n to be sufficiently big so that $\{x_t\}_{t>n}$ is sampled from a distribution that is close enough to π. The number of steps we take until we collect a sample from the chain, is called 'burn-in time'.

Definition

The ϵ-mixing time of a markov chain, is the minimal T such that, for any starting distribution $P^{(0)}$ (*i.e.* λ),

$$\mathbf{D}_{var}(P^{(T)}; \pi) \leq \epsilon,$$

where $\mathbf{D}_{var}(q, p) = \sup_x \|q(x) - p(x)\|$ is the variational distance.

Gibbs sampling

```
**Target:** to sample from $p(\mathbf{X}), \mathbf{X} = (x^1, \cdots, x^n)$
$i = 1$
**repeat**
    sample $x_t^i \sim p(x^i | x_t^1, \cdots, x_t^{i-1}, x_{t-1}^{i+1}, \cdots, x_{t-1}^n)$
    $i = i + 1$
**until** enough
```

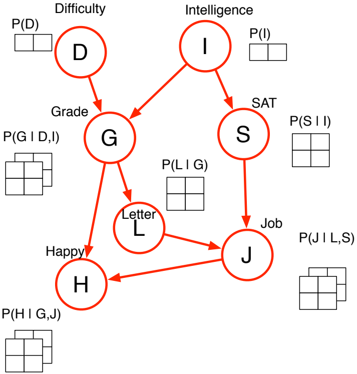
Sampling in PGM inference Overview

- Forward sampling
- Likelihood weighting sampling
- Importance sampling inference
- Gibbs sampling inference
- Metropolis-Hastings inference
- . . .

Forward sampling

Given an ordering of subsets of random variables $\{X^i\}_{i=1}^n$ (knowing parents to generate children).

```
for $i = 1$ to $n$ do
    $\mathbf{u}^i \leftarrow Pa_{\mathbf{x}^{i-1}}$
    sample $\mathbf{x}^i$ from $P(X^i | \mathbf{u}^i)$
end for
```



Forward sampling

Assume $\{\mathbf{x}_i\}_{i=1}^{M}$ are M samples from $P(X)$, we can approximately compute

- expectation:

$$\mathbb{E}_{X\sim P(X)}[f(X)] \approx \frac{1}{M}\sum_{i=1}^{M} f(\mathbf{x}_i)$$

- MAP solution: $\mathrm{argmax}_{\mathbf{x}}\, P(\mathbf{x}) \approx \mathrm{argmax}_{\mathbf{x}\in\{\mathbf{x}_i\}_{i=1}^{M}}\, P(\mathbf{x})$
- marginal: $P(\mathbf{x}) \approx N_{X=\mathbf{x}}/N_{total}$
- sample from $P(X|\,\mathbf{e})$ when evidences $\mathbf{e}$:
  sample from $P(X)$ first, and reject $\mathbf{x}$ when it does not agree on $\mathbf{e}$.

Forward sampling

Problems?

Forward sampling

Problem: Rejection step in estimating $P(X \mid \mathbf{e})$ wastes too many samples when $P(\mathbf{e})$ is small. In real applications, $P(\mathbf{e})$ is almost always very small.

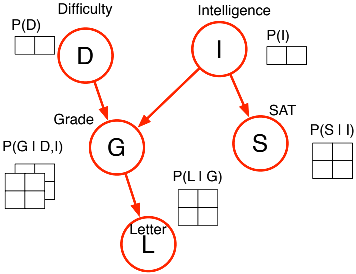
Question: how do we avoid rejecting samples?

Forward sampling

How about setting the observed random variables to the observed values, and then doing forward sampling on the rest?

Forward sampling

Let's see if it works.
To sample from $P(D, I, G, L|S = 0)$ from a simplified PGM.



Fixing $S = 0$, and then sample D, I, G, L.
Does this give the same result comparing to forward sampling with rejection?

Forward sampling

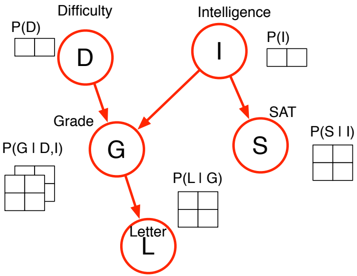
No! It doesn't.
The samples are not from $P(D, I, G, L|S = 0)$ at all!
Fixing this lead to Likelihood weighting sampling.

Likelihood weighting sampling

Input: $\{Z^i = \mathbf{z}^i\}_i$ are observed.
Step 1: set $\{Z^i\}_i$ to the observed values.
Step 2: forward sampling the unobserved variables.
Step 3: weight the sample by $\prod_i P(\mathbf{z}^i \,|\, Pa(\mathbf{z}^i))$

Likelihood weighting sampling inference

To sample from $P(D, I, G, L|S = 0)$ from the following PGM.



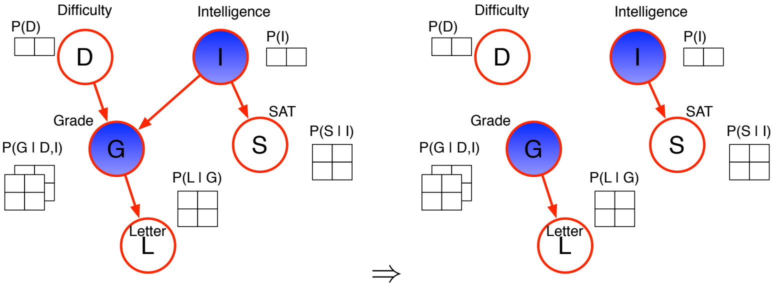
Fix $S = 0$, and forward sample D, I, G, L. Then weight the sample by $P(D, I, G, L|S = 0)$.
Does this give the same result comparing to forward sampling with rejection?

Likelihood weighting sampling

$$\mathbb{E}_{X \sim P(D,I,G,L|S=0)}[f(D,I,G,L,0)]$$

$$\approx \frac{1}{N}\sum_{j=1}^{N}[f(d_j,i_j,g_j,l_j,0)\cdot P(d_j,i_j,g_j,l_j|S=0)]$$

Importance sampling inference



(a) $\mathcal{G}_1$ with $p(X)$

(b) $\mathcal{G}_2$ with $q(X)$

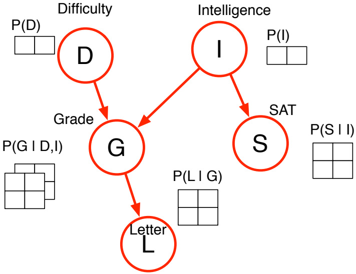
Mutilate

Sample $\{\mathbf{x}_i\}_{i=1}^{N}$ from $q(X)$.

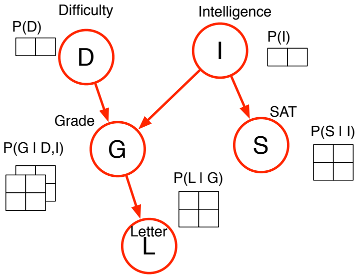
$$\hat{\mathbb{E}}_{X\sim p(X)}[f(X)] = \frac{1}{N}\sum_{i=1}^{N}\frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i)}f(\mathbf{x}_i).$$

Gibbs sampling inference

To sample from $P(D, I, G, L | S = 0)$ from the following PGM.



Gibbs sampling inference



Target: To sample $\mathbf{x} \sim P(D, I, G, L, S)$.
Given any order $\mathbf{x}^{(i)}$ (say D, I, G, S, L). Randomly initialise $\mathbf{x}$. $i = 1$

```
repeat
    sample $x^i \sim P(x^i | x^{-i})$
    $i = i + 1$
until enough
```

Gibbs sampling inference

$P(x^i|x^{-i})$ turns out easy to compute.

$$P(x^i|x^{-i}) = \frac{\prod_j P(x^j|Pa_{x^j})}{\sum_{x^i}\prod_j P(x^j|Pa_{x^j})}$$
$$= \frac{\prod_{j:x^i\in D_j}\Phi(x^i, D_j - \{x^i\})}{\sum_{x^i}\prod_{j:x^i\in D_j}\Phi(x^i, D_j - \{x^i\})}$$

Terms in which $x^i \notin D_j$ cancel out. For example,

$$x^1 = D \sim P(D|G, I, S, L) = \frac{q(D)q(G|D, I)}{\sum_D q(D)q(G|D, I)}.$$

In BN, it also turns out the only variables remaining in $P(x^i|x^{-i})$ are x^i and its Markov blanket. Similarly in MRFs.

Metropolis-Hastings inference

```
**Target:** To sample $\mathbf{x} \sim P(x)$.
Given any order $\mathbf{x}^{(i)}$. Randomly initialise $\mathbf{x}$. $i = 1$
for $t = 1, 2, \cdots$ # iterations do
    for $i = 1, 2, \cdots$ # nodes do
        Sample $x^i \sim q(x^i | x_t^i, x_t^{-i})$, $u \sim Uniform[0, 1]$
        Instead of $A(x_t \to x') = \min\{1, \frac{\pi(x')q(x_t|x')}{\pi(x_t)q(x'|x_t)}\}$
        $A(x_t^{-i}, x_t^i \to x_t^{-1}, x^i) = \min\{1, \frac{\pi(x^i, x_t^{-1})q(x_t^i|x^i, x_t^{-i})}{\pi(x_t^i, x_t^{-i})q(x^i|x_t^i, x_t^{-1})}\}$
        if $u \leq A(x_t^{-i}, x_t^i \to x_t^{-1}, x^i)$ then
            $x_t^i = x^i$;
        else
            $x_t^i = x_t^i$;
        end if
    end for
    $x_{t+1} = x_t$;
end for
```

Metropolis-Hastings inference

$\frac{\pi(x^i,x_t^{-1})q(x_t^i|x^i,x_t^{-i})}{\pi(x_t^i,x_t^{-i})q(x^i|x_t^i,x_t^{-1})}$ is easy to compute. In BN, the only variables remaining above (the rest cancels out) are x^i and its Markov blanket. Similarly in MRFs.

Again, only collect samples after burn-in time.

Next tutorial:
Temporal Models (such as models used in tracking).