PRML과 부수적인 자료들을 공부하여 간단히 정리하였습니다.

probabilistic model에서 exact inference는 intractable한 경우가 많아서 approximation한다. 이번에는 approximate inference 방법 중 *Monte carlo* 라고 알려진 numerical sampling에 기반을 하는 방법을 공부할 것이다.

우리는 posterior 자체에도 관심이 있지만 주로 expectation에 관심이 있다 (for prediction). 왜?

- expectation으로 어떤 probability도 구할 수 있다.
  - $P(X \in A) = E[I(X \in A)]$
- intractable한 sum, integral을 계산할 수 있다.

구제적으로 아래의 값을 analytical 하게 계산이 어려운 경우 Sampling을 통해 approximation한다.

$$E[f] = \int f(z)p(z)dz$$

- $p(z)$에서 iid하게 sample z_i 들을 뽑아서 평균에 대해 근사

$$\hat{f} = \frac{1}{n}\sum_{i=1}^{n} f(z_i)$$

여기서 발생 할 수 있는 문제는

- sampling한 data들이 independent 하지 않는 경우 존재
  - 그래서 effective sample size가 apparent sample size보다 훨씬 작을 수도 있다
- $p(z), f(z)$에 따라 data가 많이 필요한 경우 존재
  - normal 분포같은 경우는 괜찮은데 Gaussian mixture같이 분포가 왔다갔다하는 경우는 sample에 따라 expectation값이 천차만별

+ Monte carlo approximation

- Goal
  - approximate $E[f(X)]$
  - where $X \sim P$
- Define
  - if $X_1, X_2, ..., X_n \overset{iid}{\sim} P$
  - then $\hat{\mu}_n = \frac{1}{n}\sum f(X_i)$ is a monte carlo estimator of $E[f(X)]$
- Remark
  - $E[\hat{\mu}_n] = E[f(X)]$: unbiased estimator
  - $\hat{\mu}_n \xrightarrow{p} E[f(X)]$: consistent estimator (by WLLN)
  - $V[\hat{\mu}_n] = \frac{1}{n^2}V[\sum f(X_i)] = \frac{1}{n}V[f(X)] \to 0$

11.1 Basic Sampling Algorithms

forward sampling, regection sampling, importance sampling은 이제 별로 쓰이지 않는다고 한다. 뒤에서 배울 MCMC기법들에 조금 더 집중하자.

11.1.2 Rejection sampling

complex한 분포에서 sampling하게 도와준다.

- $p(z)$에서 바로 sampling하기는 어려운 상태
- 단, z를 넣어을 때 $p(z)$의 값은 알 수 있는 상황이다.

이를 위해 sampling이 쉬운 $q(z)$를 정한다.

- 1. $p(z)$를 다 포함하는 (envelop하는) 분포 $kq(z)$를 만든다.

$$kq(z) \geq p(z)$$

- 2. $q(z)$ 에서 sampling한다 : z_0
- 3. $unif[0, kq(z_0)]$ 에서 숫자를 generate 한다.
- 4. 해당 숫자가 $p(z_0)$ 보다 크면 reject, 아니면 sampling 한다.

따라서 $p(z)$를 잘 envelop하는 적절한 분포를 찾으면 위의 과정을 반복하여 우리가 원하는 sample들을 얻을 수 있다.

11.1.4 Importance sampling

expectation을 approximation하는 방법을 제안하다.

- where $z_i \sim p(z)$

$$E[f] \approx \frac{1}{n} \sum f(z_i)$$

하지만 $p(z)$에서 directly sampling하기 어려운 경우가 있다. so how?

- draw the samples from a proposal distribution(sampling할 수 있는), say $q(z)$
- approximation
  - where $w_i = \frac{p(z_i)}{q(z_i)}, z_i \sim q(z)$

$$E[f] = \int f(z) \frac{p(z)}{q(z)} q(z) dz \approx \frac{1}{n} \sum w_i f(z_i)$$

rejection sampling과는 다르게 sampling한 모든 data들은 버려지지 않고 사용된다.

11.1.6 Sampling and the EM algorithm

Monte carlo를 이용하여 MLE를 구할 수도 있다. EM algorithm의 E step에서 sampling methods를 통해 approximation해보자.

- complete-data log likelihood

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \int p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old}) \ln p(\mathbf{Z}, \mathbf{X}|\boldsymbol{\theta}) d\mathbf{Z}$$

- $\mathbf{Z}^{(l)}$ drawn from the current estimate for the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) \approx \frac{1}{L} \sum_{l=1}^{L} \ln p(\mathbf{Z}^{(l)}, \mathbf{X}|\boldsymbol{\theta})$$

이후에 똑같이 M-step으로 optimize한다. 이런 과정을 *Monte Carlo EM algorithm* 이라고 한다.

11.2 Markov Chain Monte Carlo

지금까지 살펴본 sampling 방법들은 high dimension에서 한계점을 갖고 있다. 이제 더 좋은 방법인 MCMC에 대해 공부해보고자 한다.

- 이전의 방법들과 마찬가지로 proposal distribution에서 sampling한다.
  - proposal distribution : $q(\mathbf{z}|\mathbf{z}^{(\tau)})$
  - 즉, current state $\mathbf{z}^{(\tau)}$ 를 given으로 하는 것이다.
  - 이런 통해 만들어진 sample들은 Markov chain을 형성한다.
- $p(\mathbf{z}) = \tilde{p}(\mathbf{z})/Z_p$ 에서 Z_p는 모르는 상태이여도 $\tilde{p}(\mathbf{z})$의 값은 구할 수 있다고 가정한다.
  - Z_p는 unknown constant
  - $p(\mathbf{z})$에서 directly sampling은 어려운 상태이다.

- proposal distribution에서 sample을 뽑고 적절한 기준으로 이를 sample로 인정할지 말지 결정한다.

위와 같은 과정을 하는 basic *Metropolis* algorithm에 대해 살펴보자.

- proposal distribution은 symmetric하게 지정한다.
  - 뒤에서 알게 되겠지만 symmetric하면 Markov chain이 time reversible하게 되고 이를 통해 stationary distribution (여기서는 $p(\mathbf{z})$ 의미) 이 존재한다.

$$q(\mathbf{z}_A|\mathbf{z}_B) = q(\mathbf{z}_B|\mathbf{z}_A)$$

- 확률적으로 sample로 인정한다, 아래의 식이 accept 확률이다.
  - 딱 봤을때 적절하다는 생각이 든다. 이렇게 반복하다 보면 우리가 원하는 $p(\mathbf{z})$의 모양으로 sampling이 될 것이다.
  - $\frac{p(\mathbf{z}^*)}{p(\mathbf{z}^{(\tau)})} = \frac{\tilde{p}(\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})}\frac{Z_p}{Z_p} = \frac{\tilde{p}(\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})}$ 라서 정확한 $p(\mathbf{z})$의 값을 몰라도 합리적인 acceptance probability를 구할 수 있다.

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = min(1, \frac{\tilde{p}(\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})})$$

- unif(0,1)에서 random number u를 뽑는다. $A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) > u$이면 sample을 accept한다.

candidate sample이 accpet되면 그 sample을 sample list에 저장한다. 그리고 그 sample을 given한 상태로 다시 알고리즘을 진행한다. 만약 reject되면 그 때 당시의 given sample을 sample list에 추가하고 다시 알고리즘을 진행한다. 각 sequence sample은 independent한 sample이 아니다. highly correlated되어 있다면 sample list에서 띄엄띄엄 사용하면 된다.

11.2.1 Markov chains

대표적인 MCMC 알고리즘을 살펴보기 전에 Markov chain에 대해 공부해보자.

각 state에 해당하는 random variable $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(M)}$ 가 있다. 이들이 아래와 같은 conditional independence property를 갖을 때, 이러한 stochastic process를 Markov chain이라고 한다.

$$p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(m)}) = p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)}), m \in \{1, .., M-1\}$$

- transition probability

$$T(\mathbf{z}^{(m)}, \mathbf{z}^{(m+1)}) \equiv p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)}) \equiv T_{\mathbf{z}^{(m)}, \mathbf{z}^{(m+1)}}$$

이제 Markov chain의 몇 가지 특징들에 대해 살펴보자.

- **accesible**
  - $i \to j$: state j is *accessible* from i if $T_{i,j}^m > 0$
    - 즉 state i에서 언젠가는 state j에 방문한다는 의미
  - $i \leftrightarrow j$: state i, j *communicate* if $i \to j$ and $j \to i$
- **reducibility**
  - 모든 state i, j가 communicate하다면 그 Markov chain은 *irreducible* 하다고 한다.
- **periodicity**
  - state i has *peoriod d* if $d = gcd\{n : T_{i,j}^m > 0\}$
  - if d = 1, state i is *aperiodic*
- **transience**
  - state j is *recurrent* if j에서 시작해서 언젠가는 다시 j를 방문할 확률이 1인 경우
  - state which is not recurrent is *transient*
  - *positive recurrent* : expected time until the process starting in state i returns to i is finite
- **ergodicity**
  - irreducible, aperiodic, positive recurrent Markov chain on a countable state space is called *ergodic*
    - 참고로 irreducible가 성립하면 자동으로 positive recurrent가 성립한다.
    - 따라서 어떤 책에는 ergodic의 조건으로 irreducible, aperiodic 만을 언급하기도 한다.
    - countable state space가 아닌 general한 경우는 ergodicity를 확인하기 복잡하다.

        - (어떤 책에는) For countable state spaces, an irreducible, aperiodic Markov chain having a stationary distribution is ergodic.
    - ergodic Markov chain은 unique stationary distribution을 갖고 있다.
- **Stationary distribution**
    - regular MC는 limiting probability distribution $\boldsymbol{\pi} = (\pi_0, ..., \pi_N)$을 갖는다.
        - transition matrix $\mathbf{T}^{(m)}$의 모든 원소가 0보다 크면 MC가 *regular* 하다고 한다.
        - $\pi_j = lim_{n\to\infty} T_{i,j}^{(n)}$
    - stationary distribution은 존재하지 않을 수도 있고 여러 개일 수도 있다.
    - 아래와 같은 식을 만족하는 limiting distribution을 stationary distribution이라고 부른다.

$$\pi_j = \sum_{k=0}^{K} \pi_k T_{k,j}$$

- **detailed balance condition**
    - Markov chain이 아래와 같은 식을 만족하면 *time reversible* 하다고 한다.
        - $P(\mathbf{z}^{(n)} = j | \mathbf{z}^{(n+1)} = i) = P(\mathbf{z}^{(n+1)} = j | \mathbf{z}^{(n)} = i)$
    - time reversibility의 조건은 아래와 같이도 표현할 수 있는데 이를 *detailed balance condition* 이라고 한다. (detailed balance condition $\Leftrightarrow$ reversibility)

$$\pi_i T_{i,j} = \pi_j T_{j,i}$$

- detailed balance condition을 만족하면 stationary distribution을 갖는다. (unique한지는 확신할 수 없다)
    - proof

$$\begin{aligned}
\pi_i T_{i,j} &= \pi_j T_{j,i} \\
\sum_i \pi_i T_{i,j} &= \sum_i \pi_j T_{j,i} \\
\sum_i \pi_i T_{i,j} &= \pi_j \sum_i T_{j,i} \\
\sum_i \pi_i T_{i,j} &= \pi_j
\end{aligned}$$

지금까지 Markov chain에 대해 알아보았다. 이 특성들을 이용하여 우리는 MCMC algorithm을 진행한다. 일단 전통적인 Markov chain 이론과 MCMC의 구별되는 특징에 대해 알아보면

- In the traditional Markov chain theory,
    - Given a transition rule, $P(\mathbf{z}^{n+1} = j | \mathbf{z}^{(n)} = i)$
    - Interested in finding its stationary distribution π
- In the MCMC,
    - Given a target stationary distribution π
    - Interested in prescribing and efficient transition rule to reach π

이 내용을 위에서 봤던 Metropolis algorithm과 잘 연결시켜서 이해하도록 하자.

- ergodic MC는 unique stationary distribution을 갖고 있다. 하지만 종종 ergodic 여부를 판단하기 어려운 상황이 발생한다.
- in practice, reversible MC는 detailed balance condition을 만족하고 이는 stationary distribution가 존재함을 위미한다. 따라서 reversible MC를 주로 이용하고 starting value를 여러가지로 진행하여 unique함을 확인한다.

11.2.2 The Metropolis-Hastings algorithm

이전에 Metropolis algorithm에서는 proposal distribution(= transition kernel)이 symmetric했다. 하지만 이제는 그렇지 않다. 단, $q(\mathbf{z}_A | \mathbf{z}_B) > 0 \Leftrightarrow q(\mathbf{z}_B | \mathbf{z}_A) > 0$ 을 만족해야 한다.

- 현재 state는 $\mathbf{z}^{(\tau)}$
- distribution $q(\mathbf{z} | \mathbf{z}^{(\tau)})$로부터 sample $\mathbf{z}^*$을 뽑는다.
    - normal 분포를 주로 사용한다. 단, 분산을 적절하게 선택해야 한다.

- 아래의 확률에 따라 accept한다.

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = min(1, \frac{\tilde{p}(\mathbf{z}^*)q(\mathbf{z}^{(\tau)}|\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})q(\mathbf{z}^*|\mathbf{z}^{(\tau)})})$$

- Metropolis-Hastings은 detailed balance condition을 만족한다.
  - proof

$$\begin{aligned} p(\mathbf{z})T_{\mathbf{z},\mathbf{z}'} = p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})A(\mathbf{z}', \mathbf{z}) \\ = min\{p(\mathbf{z})q(\mathbf{z}'|\mathbf{z}), p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')\} \\ = min\{p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}'), p(\mathbf{z})q(\mathbf{z}'|\mathbf{z})\} \\ = p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}')A(\mathbf{z}, \mathbf{z}') = p(\mathbf{z}')T_{\mathbf{z}',\mathbf{z}} \end{aligned}$$

detailed balance condition을 만족하기에 우리가 sampling하여 만들어지는 MC가 stationary distribution을 갖게 된다. 따라서 stationary distribution으로 수렴하기전의 sampling 초반의 sample들은 버리고 (해당 구간을 burn-in period 라고 한다) 뒷부분의 sample들을 이용한다. 해당 sample list는 stationary distribution의 형태를 갖고 있을 것이다.

11.3 Gibbs sampling

Metropolis-Hastings algorithm의 특별한 케이스이다. 우리가 sample을 뽑고 싶어하는 $p(\mathbf{z}) = p(z_1, z_1, ..., z_M)$ distribution이 있다. 이전에 우리는 proposal distribution을 따로 정해서 사용하였지만 Gibbs sampling에서는 그렇지 않다. 먼저 1부터 순서대로 z_i를 distribution $p(z_i/\mathbf{z}_{-i})$에서 뽑는다. 이를 M까지 반복한다.

- Gibbs sampling
  - Innitialize $\{z_i : i = 1, ..., M\}$
  - For $\tau = 1, ..., T$:
    - Sample $z_1^{(\tau+1)} \sim p(z_1|z_2^{(\tau)}, z_3^{(\tau)}, ..., z_M^{(\tau)})$
    - Sample $z_2^{(\tau+1)} \sim p(z_2|z_1^{(\tau+1)}, z_3^{(\tau)}, ..., z_M^{(\tau)})$
    - ...
    - Sample $z_M^{(\tau+1)} \sim p(z_M|z_1^{(\tau+1)}, z_2^{(\tau+1)}, ..., z_{M-1}^{(\tau+1)})$

즉, Gibbs sampling에서는 acceptance probability를 사용하지 않는다. 모든 sample을 그대로 사용한다. 그렇게 해도 detailed balance condition을 만족하는지 살펴보자.

- Gibbs sampling에서 $q(\mathbf{z}/\mathbf{z}') = p(z_i/\mathbf{z}'_{-i})$ 이고
- $p(\mathbf{z}')q(\mathbf{z}/\mathbf{z}') = p(\mathbf{z})q(\mathbf{z}'/\mathbf{z})$ 임을 확인해보자.

$$\begin{aligned} p(\mathbf{z}')q(\mathbf{z}|\mathbf{z}') = p(z_i', \mathbf{z}'_{-i})p(z_i|\mathbf{z}'_{-i}) \\ = p(z_i'|\mathbf{z}'_{-i})p(\mathbf{z}'_{-i})p(z_i|\mathbf{z}'_{-i}) \\ = p(z_i'|\mathbf{z}'_{-i})p(z_i, \mathbf{z}'_{-i}) = q(\mathbf{z}'|\mathbf{z})p(\mathbf{z}) \end{aligned}$$

항상 detailed balance condition이 성립한다. 또한 acceptance probability도 1이 된다. sample을 항상 accept한다.