Gibbs Sampler

ESC-21WINTER : Bayes 스터디 소개

0. Recap

Bayes rule $\quad p(\theta|D) = \dfrac{p(D|\theta)p(\theta)}{\int p(D|\theta)p(\theta)d\theta}$

- conjugate prior
- sampling

1. Univariate Normal model (unknown μ)

- Prior $\quad \mu \sim N(\mu_0, \tau_0^2)$
- Likelihood $\quad y|\mu, \sigma^2 \sim N(\mu, \sigma^2)$
- Posterior $\quad \mu|y, \sigma^2 \sim N(\mu_n, \tau_n^2)$

  where $\quad \mu_n = \dfrac{1/\tau_0^2}{n/\sigma^2 + 1/\tau_0^2}\mu_0 + \dfrac{n/\sigma^2}{n/\sigma^2 + 1/\tau_0^2}\bar{y} \quad$ and $\quad \tau_n = \dfrac{1}{n/\sigma^2 + 1/\tau_0^2}$

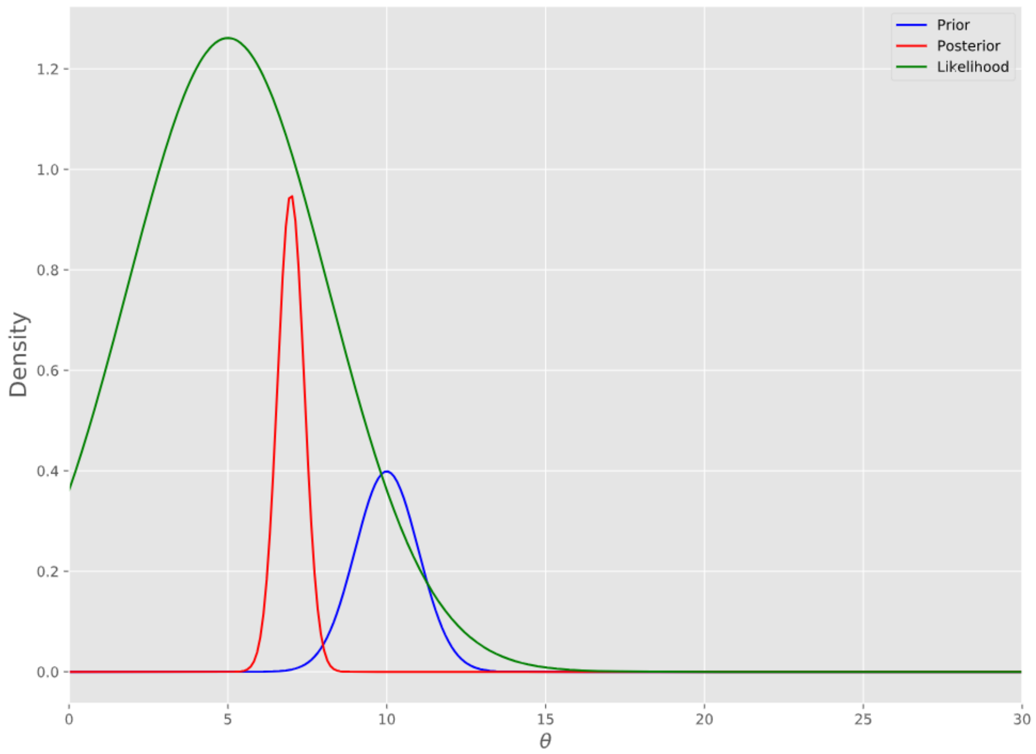
Simulation

```
# Likelihood
n = 10
mu = 5
tau = np.sqrt(10)
rv = st.norm.rvs(size=n, loc=mu, scale=tau)
ybar = rv.mean()
sigma = rv.std()

# Prior
mu0, tau0 = 10, 1
prior = st.norm(loc=mu0, scale=tau0)

# Posterior
mu_n = ((1/tau0**2)/(n/sigma**2 + 1/tau0**2))*mu0 + ((n/sigma**2)/(n/sigma**2 + 1/tau0**2))*ybar
tau_n = 1/(n/sigma**2 + 1/tau0**2)
post = st.norm(mu_n, tau_n)
```

- Prior $\mu \sim N(\mu_0, \tau_0^2)$
- Likelihood $y|\mu, \sigma^2 \sim N(\mu, \sigma^2)$
- Posterior $\mu|y, \sigma^2 \sim N(\mu_n, \tau_n^2)$

  where $\mu_n = \dfrac{1/\tau_0^2}{n/\sigma^2 + 1/\tau_0^2}\mu_0 + \dfrac{n/\sigma^2}{n/\sigma^2 + 1/\tau_0^2}\bar{y}$ and $\tau_n = \dfrac{1}{n/\sigma^2 + 1/\tau_0^2}$

```
# plotting
thetas = np.linspace(0, 30, 300)
plt.figure(figsize=(12, 9))
plt.style.use('ggplot')
plt.plot(thetas, prior.pdf(thetas), label='Prior', c='blue')
plt.plot(thetas, post.pdf(thetas), label='Posterior', c='red')
plt.plot(thetas, n*st.norm(mu, tau).pdf(thetas), label='Likelihood', c='green')
plt.xlim([0, 30])
plt.xlabel(r'$\theta$', fontsize=14)
plt.ylabel('Density', fontsize=16)
plt.legend();
```



" We want to know $p(\mu, \Sigma \mid y)$ "

2. Multivariate Normal model : semi-conjugacy

MVN distribution $\quad y \sim N(\mu, \Sigma) = N(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix})$

conjugate prior

$\begin{cases} \mu \\ \mu|y, \Sigma \end{cases}$ $\quad \begin{cases} \Sigma \\ \Sigma|y, \mu \end{cases}$ $\quad \cancel{\begin{cases} \mu, \Sigma \\ \mu, \Sigma|y \end{cases}}$ Not a conjugate prior!

known Cov. matrix

$$\begin{cases} \mu \sim N(\mu_0, \Lambda_0) \\ y \sim N(\mu, \Sigma) \\ \mu|y, \Sigma \sim N(\mu_n, \Lambda_n) \end{cases}$$

known mean vector

$$\begin{cases} \Sigma \sim Wis^{-1}(\nu_0, S_0^{-1}) \\ y \sim Wis^{-1}(n, S_\mu^{-1}) \\ \Sigma|y, \mu \sim Wis^{-1}(\nu_0 + n, (S_0 + S_\mu)^{-1}) \end{cases}$$

Now we know full-conditional distributions!

$$p(\theta_i|\theta_1, \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_n) = p(\theta_i|\theta_{-i}) \;\; for \;\; \forall \theta_i$$

3. Gibbs Sampling

우리가 알고 싶은 분포는 $p(\mu, \Sigma|y)$ 이지만 많은 경우 이런 joint conditional distribution의 closed form을 바로 구하기가 어렵다.

이런 상황에서 우리가 full conditional distribution, 즉 $p(\mu|y, \Sigma)$ 와 $p(\Sigma|y, \mu)$ 는 알고 있다면 이들로부터 $p(\mu, \Sigma|y)$ 를 approximation할 수 있다.

< Algorithm >

1. Initialize $\mu^{(1)}, \Sigma^{(1)}$
2. sample $\mu^{(2)} \sim p(\mu^{(1)}|\Sigma^{(1)})$
3. sample $\Sigma^{(2)} \sim p(\Sigma^{(1)}|\mu^{(2)})$
4. Repeat 2-3 for n times!

이렇게 sampling 과정을 무수히 많이 반복하면 우리는 $\mu = [\ \mu^{(1)},\ \mu^{(2)}, \ldots,\ \mu^{(n)}],\ \Sigma = [\ \Sigma^{(1)},\ \Sigma^{(2)}, \ldots,\ \Sigma^{(n)}]$ 가 $p(\mu, \Sigma|y)$로부터 왔다고 볼 수 있다.

정리하자면 Gibbs Sampling method는

- sampling을 통해 target distribution (posterior)를 복원, 또는 재현하는 메커니즘이다.

- multi-dimension problem을 single-dimension problem으로 바꿔서 연산을 간소화한다. (single-dimension sampling!)

4. Applications of Gibbs Sampling

- LDA : 모델 parameter 추정

- Image Denoising : recovering corruped image

- NA imputation

Simulation

Case 1 : Sampling μ, Σ from full probability distribution

$< y \mid \mu, \Sigma$

```
# Prior    μ에 대한 prior    μ ~ N(μ0, Λ0)
mu0 <- c(50,50)
L0 <- matrix( c(625,312.5,312.5,625), nrow=2, ncol=2) # lambda0

# Hyperparameters    Σ 에 대한 prior    Σ ~ Wis^-1(ν0, S0^-1)
nu0 <- 4  # number of times to sample
S0 <- (nu0 - nrow(L0) - 1) * L0 # form of cov matrix

# Gibbs Sampler
inv = solve
n = nrow(Y)
ybar = colMeans(Y)
Sigma = cov(Y) # initialization s(1)
S = 5000
MU = matrix(NA, nrow=S, ncol=2)
SIGMA = matrix(NA, nrow=S, ncol=4)
YS <- NULL

for (s in 1:S) {  # full conditional posterior
  # update mu   draw μ from p(μ|Σ,y)
  Ln = inv(inv(L0) + n*inv(Sigma))
  mun = Ln %*% (inv(L0) %*% mu0 + n*inv(Sigma) %*% ybar)
  mu = MASS::mvrnorm(n=1, mun, Ln)

  # update sigma   draw Σ from p(Σ|μ,y)
```

```
  Sn = S0 + (t(Y) - c(mu)) %*% t((t(Y)-c(mu)))
  Sigma = inv(rWishart(1, nu0+n, inv(Sn))[,,1])
  YS<-rbind(YS,rmvnorm(1,theta,Sigma))  # sample (y1, y2) from updated joint dist (MVN)

  MU[s,] = mu
  SIGMA[s,] = c(Sigma)
}
```

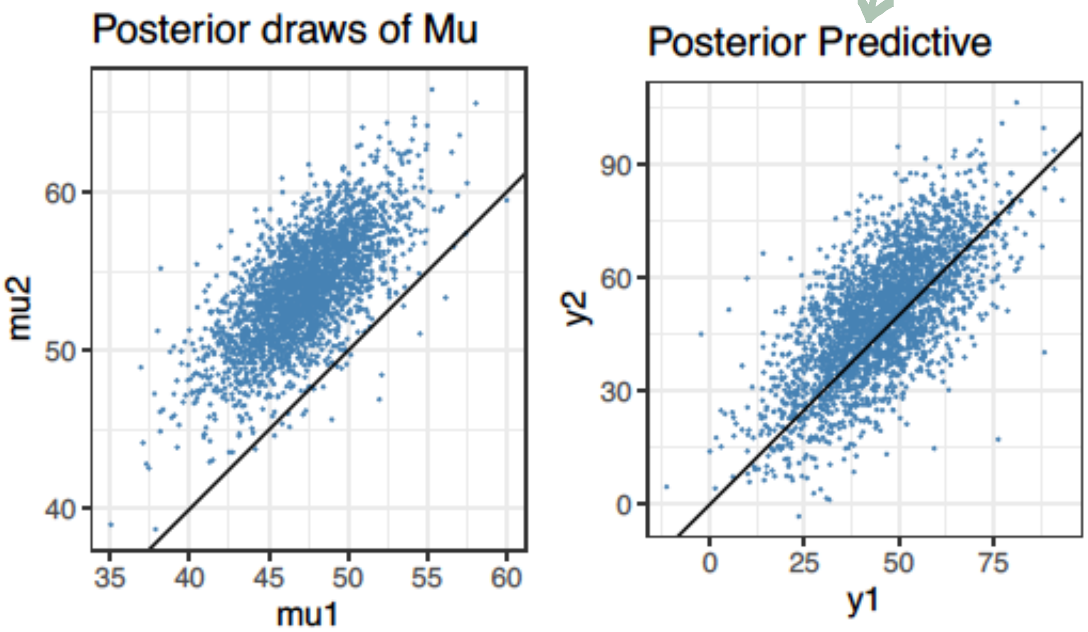
(오른쪽 plot)

μ, Σ 에 대한 추론을 할 수 있게됨

ex. $p(\mu_1 < \mu_2 \mid y)$

다음의 예시는 μ, Σ 를 full-prob dist.에서 sampling 한 뒤 MU 의 tuple (μ_1, μ_2)을 scatterplot 형태로 찍은 것

(왼쪽 plot)



Case 2 : Sampling $[x_1, x_2]$ directly from known "complicated" distribution

Gibbs Sampler는 Case 1에서처럼 분포 자체를 추정할 때도 사용되지만, high-dimensional joint distribution에서 직접 샘플을 뽑는 게 어려울 경우 하나의 차원 단위로 sampling을 하여 표본을 얻는데도 사용될 수 있다.

예를 들어, 다음과 같은 Bivariate Normal 분포에서 (x1, x2)를 sampling하고싶다고 하자.

(물론 bivariate normal은 직접 sampling 하는 것이 어렵지는 않지만 Gibbs Sampler가 어떻게 동작하는지 시각화하기 위해 아래 예시를 채택하였음!)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim N(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix})$$

이 때 하나의 차원 단위로 sampling 한다는 것은 $p(x_1|x_2)$로부터 x1을 뽑고, $p(x_2|x_1)$로부터 x2를 뽑는 것을 반복하여 sample을 generate 한다고 이해하면 된다.

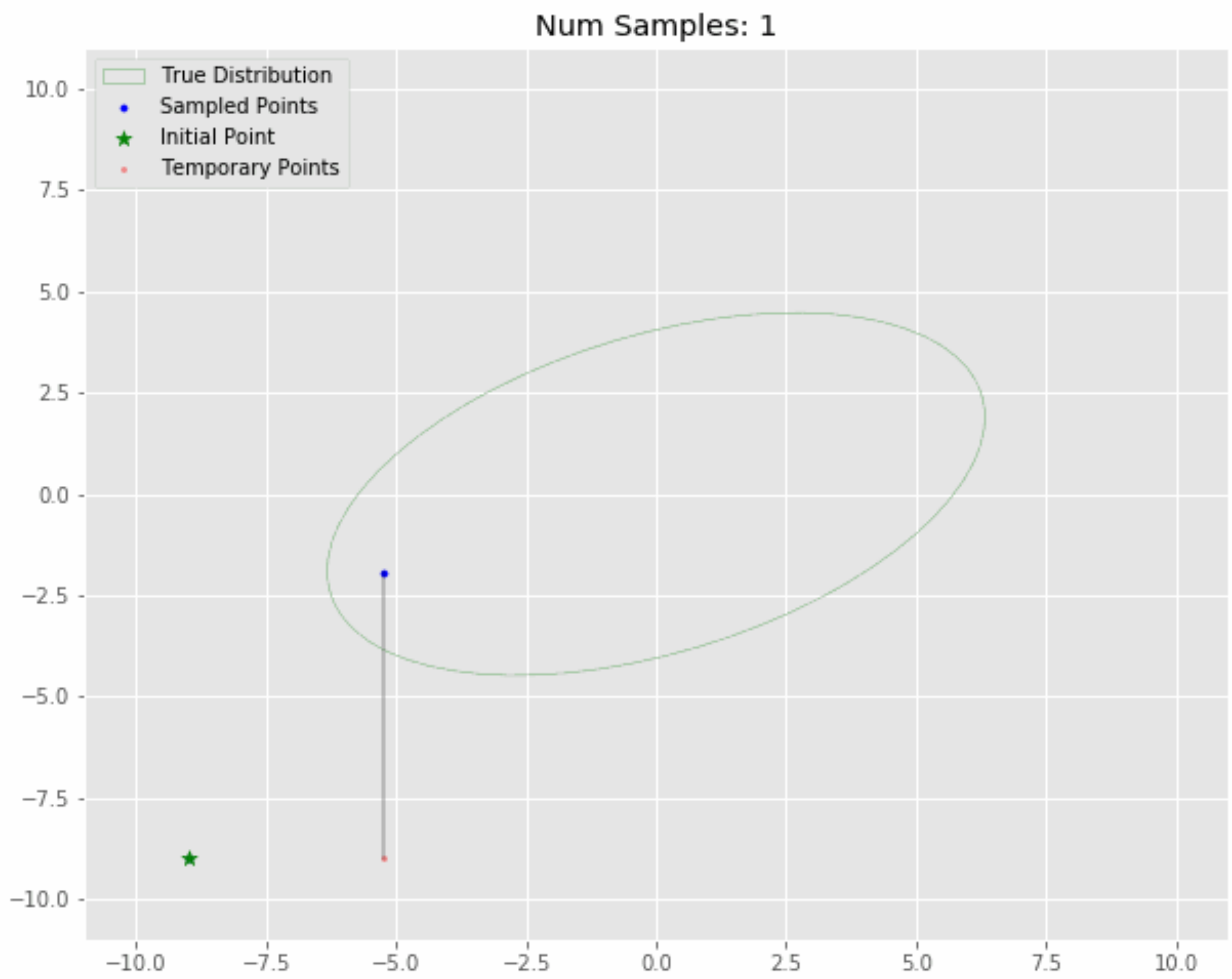
```
# pseudo-code
def gibbs_sampler(initial_point, num_samples, ...):
    x_1, x_2 = initial_point[0], x_2 = initial_point[1]
    samples = np.empty([num_samples+1, 2])  #sampled points
    samples[0] = [x_1, x_2]

    for i in range(num_samples):
        # Sample from p(x_1|x_2)
        x_1 = conditional_sampler(sampling_index=1, condition_on=x_2, ...)
        # Sample from p(x_2|x_1)
        x_2 = conditional_sampler(sampling_index=2, condition_on=x_1, ...)
        samples[i+1] = [x_1, x_2]

    return samples
```

그리고 $p(x_1|x_2) \sim N(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), \Sigma_{11} - \frac{\Sigma_{12}^2}{\Sigma_{22}})$ 임을 이용하여 conditional sampler를 다음과 같이 구현할 수 있다.

```
def conditional_sampler(sampling_index, current_x, mean, cov):
    conditioned_index = 1 - sampling_index
    a = cov[sampling_index, sampling_index]
    b = cov[sampling_index, conditioned_index]
    c = cov[conditioned_index, conditioned_index]

    mu = mean[sampling_index] + (b * (current_x[conditioned_index] - mean[conditioned_index]))/c
    sigma = np.sqrt(a-(b**2)/c)
    new_x = np.copy(current_x)
    new_x[sampling_index] = np.random.randn()*sigma + mu

    return new_x
```

이제 μ, Σ가 주어진 상황에서 Gibbs Sampler를 이용해 표본을 추출할 때 원래의 모분포에 점점 수렴하는 방식으로 scatter plot이 찍히고 ellipse가 변화하는 것을 볼 수 있다.

```
# original distribution
mean = np.array([0, 0])
cov = np.array([[10, 3],
                [3, 5]])

initial_point = [-9.0, -9.0]
num_samples = 500
samples, tmp_points, frames = gibbs_sampler(initial_point, num_samples, mean, cov, create_gif=True)
```



Num Samples: 1
True Distribution
Sampled Points
Initial Point
Temporary Points
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0