2018 FALL
COMPUTATIONAL STATISTICS
FINAL PROJECT

Sequential Importance Sampling (SIS)

- with Applications to Discrete Time Random Walks -

과목명 | 통계계산특론1

교수명 | 송종우 교수님

제출일 | 2018년 12월 10일

학 번 | 182STG18

이 름 | 이하경

1. INTRODUCTION

1-1. DESCRIPTION

Monte-Carlo Simulation은 Target 분포 f에 대해 실제와 동일하거나 유사한 분포로부터 서로 독립적인 Random Sample을 구성한 후 평균을 이용하여 관심 함수에 대한 값을 추정하는 방법이다. 그 중 Approximate Simulation 방법 중 하나인 Sampling Importance Resampling(SIR)은 실제 분포로부터 직접적으로 Sample을 구성하기 어려운 경우, 가능한 한 실제 분포와 유사하며 Sample Draw가 쉬운 Importance Sampling Function (envelope)로부터 특정 크기의 sample을 구성하고 각각에 대한 Importance Weight $w(x) = f(x)/g(x)$을 사용하여 해당 sample로부터 원하는 크기의 sample을 다시 추출하는 방법이다. SIR을 효과적으로 이용하기 위해서는 envelope로 사용하는 g를 적절히 설정하는 것이 실제 분포 추정에 매우 중요하며, 적절한 envelope g는 다음의 조건을 만족해야 한다.

1) g의 support는 실제 분포 f의 support를 전부 포함해야 한다.
2) g는 f에 비해 두꺼운 꼬리를 가져야 한다.

위의 조건을 만족하는 분포를 이용하여 실제와 유사하며 서로 독립적인 sample X를 구성하게 되면 이를 이용해 관심함수에 대해 기댓값 E[h(X)]에 대한 Simple MC, IS(Importance Sampling) 추정치 등을 계산할 수 있다.

그러나 SIR은 추정하고자 하는 Target density f가 high-dimension일 때 수행하기에 많은 어려움이 있다. 포함된 변수가 많을 때 이를 모두 포함하는 joint distribution으로서 유사한 envelope를 찾는 것은 상당히 어렵다. 앞으로 소개할 Sequential Monte-Carlo 방법은 이러한 high-dimension에서 발생하는 어려움을 단계적인 sequence로 분리하고, 이전 sequence를 이용해 다음 sequence를 추정함으로써 해결하는 방법으로 Time Stochastic Process와 같이 특정한 시점의 Random Variable $\mathrm{X_t}$가 이전 시점까지의 모든 sequence에 대한 정보를 포함할 때 효과적으로 Sampling을 가능하게 한다. 따라서 단계적, 순차적인 추정이 실시간으로 이루어지는 사물의 궤도 추정(Object Tracking), 고분자 화합물에 대한 구조 추정 등에서 중요하게 다뤄지고 있다.

본 프로젝트에서는 Sequential Importance Sampling (with Resampling)의 개념에 대해 간단히 소개하고, SISR과 MCMC을 결합한 Resample-Move 방법을 Object Tracking Problem에 적용해보려고 한다.

1-2. Sequential Importance Sampling (SIS)

1-2-1. SIS for Markov or Non-Markov Process

Sequential IS는 기본적으로 Importance Sampling을 기반으로 한 방법이다. $\mathrm{X_{1:t}} = (\mathrm{X_1}, \dots, \mathrm{X_t})$의 time process를 이용해 $\mathrm{h(X_{1:t})}$을 추정하고자 할 때, IS 방법을 단순히 적용한다면 joint envelope $\mathrm{g_t}$로부터 $\mathrm{x_{1:t}}$의 sample을 각각 생성하고 가중치를 계산하여 추정에 이용한다. 그러나 각각의 t시점마다 사전정보 없이 sample을 획득하여 그대로 이용하는 것보다 t가 증가함에 따라 추가되는 정보를 이용해 기존의 추론을 update하는 것이 더 효율적인 방법임을 예상할 수 있다. 따라서 각 시점에서 $\mathrm{X_{1:t-1}}$의 sample set에 simulation을 통해 새롭게 얻은 $\mathrm{X_t}$를 추가하고, 이전의 가중치 또한 update하여 사용할 수 있다. 이것을 Sequential Importance Sampling이라고 한다.

$\mathrm{X_1}, \dots, \mathrm{X_t}$의 joint distribution $\mathrm{f_t}$와 $\mathrm{g_t}$는 각각 다음과 같이 표현될 수 있다.

$$f(x_{1:t}) = f(x_1)f(x_2|x_1)\cdots f(x_t|x_{1:t-1}) \quad \& \quad g(x_{1:t}) = g(x_1)g(x_2|x_1)\cdots g(x_t|x_{1:t-1})$$

또한 f와 g를 이용한 가중치 $w^*(x_{1:t})$ 또한 위의 성질을 이용하여 다음과 같이 분할될 수 있다.

$$\mathrm{w_t} = u_1 \cdot u_2 \cdots u_{t-1} \cdot u_t, \qquad u_i = f(x_i|x_{1:i-1})/g(x_i|x_{1:i-1}) \quad \rightarrow \quad w_t = w_{t-1} \cdot u_t$$

따라서 $\mathrm{X_{1:t-1}}$의 sample에 $\mathrm{X_t}$을 포함시킴과 함께 이전의 가중치에 새로운 가중치를 곱하여 계속해서 update한다. t시점마다 n개의 $\mathrm{X_t}$ 의 sample을 batch로 생성하여 관심함수에 대해 $\mathrm{h(X_t)} = \sum_{i=1}^{n} h(X_t^{(i)})/n$ (MC), 또는 $\mathrm{h(X_t)} = \sum_{i=1}^{n} w^*(X_t^{(i)})\, h(X_t^{(i)}) / \sum_{i=1}^{n} w^*(X_t^{(i)})$ (IS) 추정치를 계산할 수 있다. 만약 f와 g가 Markov process라면 t시점의 모든 함숫값과 가중치는 바로 이전 시점에만 영향을 받으므로 더욱 간단해진다.

1-2-2. SIS for Hidden Markov Models

관측이 불가능한 Markov sequence $X_0, X_1, \dots, X_t$가 있다고 하자. 각 시점의 X_t의 분포는 Markov 성질에 의해 X_{t-1}에만 영향을 받는다. 이와 더불어 각 시점의 X_t의 영향을 받는 관측이 가능한 $Y_0, Y_1, \dots, Y_t$가 존재할 때,

$$Y_t \sim f_y(y_t|x_t) \quad \& \quad X_t \sim f_x(x_t|x_{t-1})$$

와 같은 구조를 이루는 것을 Hidden Markov Process라고 한다. 따라서 $X_{1:t}$ 추정하기 위해 관측된 데이터 $y_{1:t}$를 이용하고자 할 때, target posterior density $f_t(x_{1:t}|y_{1:t})$는 다음과 같이 표현된다.

$$f_t(x_{1:t}|y_{1:t}) = f_t(x_{1:t-1}|y_{1:t-1}) \cdot f_x(x_t|x_{t-1}) f_y(y_t|x_t)$$

Markov 성질을 이용해 $f_x(x_t|x_{t-1})$을 Importance Sampling의 envelope로 사용할 경우 t시점마다 update되는 u_t는 $\frac{f_t(x_{1:t}|y_{1:t})}{f_t(x_{1:t-1}|y_{1:t-1}) f_x(x_t|x_{t-1})} = p_y(y_t|x_t)$, 즉 likelihood에 비례한다.

Bayesian framework에서 $X_{1:t}$의 joint prior distribution은 $p_x(x_0) \prod_{i=1}^{t} p_x(x_i|x_{i-1})$이며 likelihood는 $\prod_{i=0}^{t} p_y(y_i|x_i)$으로 posterior는 이 두가지의 곱에 비례한다.

따라서 t시점의 Importance weight를 t시점에 새롭게 관측된 데이터 yt의 likelihhood를 sample의 xt를 이용해 각각 계산하고 update할 수 있다. 이러한 Hidden Markov Model에 Sequential Importance Sampling을 적용하여 직접 Sampling 및 Resampling을 구현해보았다.

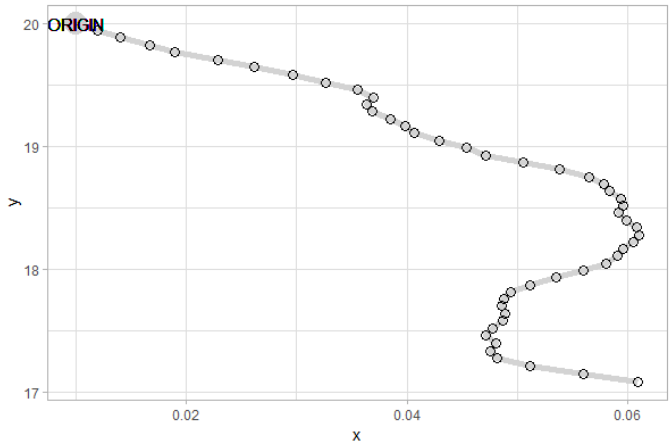
1-2-3. SIS with Resampling

SIS 역시 적절한 Importance Function g의 선택이 추정에 많은 영향을 미친다. 이와 더불어 SIS의 Performance에 영향을 크게 미치는 문제점인 Weight Degeneracy는 t가 계속해서 증가함에 따라 가중치를 계속해서 곱하여 이용할 경우 전체 가중치가 일반성을 잃어 추정에 문제를 일으키는 것을 말한다. 만약 t시점마다 n개의 sample sequence를 고려한다고 할 때 하나의 sequence에 일반적이지 않은(희귀한) sample이 포함될 경우 해당 sequence에 대한 t시점의 가중치가 낮아져 이후 계속해서 가중치가 약화되고, n개중 특정한 sequence들에만 가중치가 집중될 가능성이 있다. 이러한 문제점을 해결하기 위한 sequence들은 적절히 가중치를 '회복(rejuvenate)'해야 한다. 이를 위해서 적절한 타이밍에 해당 시점까지의 가중치를 이용하여 sequence들을 재구성한 뒤 모든 가중치를 동일하게 초기화시키는 방법을 이용한다. 이것을 Sequential Importance Sampling with Resampling이라고 한다. 그 중 하나의 예시로 Bootstap Filter란 매 시점마다 Resampling을 진행하는 방법이다. n개의 sequence에서 계산된 가중치를 확률로 다시 n개의 sample sequence을 비복원추출하고 선택되지 않은 sequence들은 버려진다. 가중치는 다음 시점으로 넘어가기 전 모두 동일하게 reset된다.

2. IMPLEMENTATION

2-1. OBJECT TRACKING PROBLEM

베이지안 추론 및 예측에서 one-step transition probability를 이용한 MCMC의 일종으로서 Gibbs Sampling은 Target distribution이 High-dimensional인 경우 매우 효과적이다. 그러나 시점마다 데이터의 누적과 더불어 target posterior distribution에서 추정해야 하는 모수가 계속해서 증가할수록 전체 모수에 대한 MCMC sampling은 많은 시간이 소요되어 사물의 실시간 궤도를 추정하는 Moving target traking과 같은 문제에서 빠르게 이용하기 어려운 점이 있다. 또다른 시뮬레이션 방법으로 사용 가능한 SIS with Resampling은 target sequence가 증가할수록 degeneration에 빠지기 쉽다는 단점이 있다. 따라서 이 두 가지를 적절히 조합한 방법으로서 계산의 부담을 줄이면서 효과적으로 추정이 가능한 방법으로 Resample-move algorithm이 제시되었다[1] (Walter R. Gilks and Carlo Berzuini, 2001).



이를 적용하는 예시인 bearing-only tracking으로서 다음과 같이 시간에 따라 random한 속도로 움직이는 선박의 관측을 고려한다. 관측지점으로부터 동쪽, 북쪽으로의 선박의 실제 위치를 x_t, y_t, 각 방향으로의 속도를 $\dot{x}_t$, $\dot{y}_t$라고 할 때 이 수치들은 관측 불가능하며 관측지점으로부터 기계의 noise를 포함한 각도 z_t만이 관측된다. 시점 t에 따른 선박의 궤도에 대한 모형은 다음과 같다.

$$\dot{x}_t = N(\dot{x}_{t-1}, \tau^{-1}), \qquad \dot{y}_t = N(\dot{y}_{t-1}, \tau^{-1})$$

$$x_t = x_{t-1} + \dot{x}_{t-1}, \qquad y_t = y_{t-1} + \dot{y}_{t-1}$$

x 방향과 y 방향으로의 속도는 알려지지 않은 모수 τ^{-1}을 분산으로 가지는 정규분포를 따르며, t시점의 x축, y축 위치는 이전 시점의 위치에 각 방향으로의 속도를 더한 값이다. 관측된 각도는 $z_t = \tan^{-1}(y_t/x_t) + N(0, \eta^2)$으로 기계장치의 random noise를 포함한다. 따라서 시점 t에 따라 새로운 z_t가 관측됨과 함께 새로운 parameter $\dot{x}_t$, $\dot{y}_t$ 가 모형에 포함된다. 따라서 t시점에 추정해야 할 모수 $\theta_t = (\tau^{-1}, x_1, y_1, \dot{x}_1, \dot{y}_1, \ldots, \dot{x}_t, \dot{y}_t)$ 는 시간에 따라 증가한다. z_t의 누적에 따라 unknown hyperparameter τ^{-1}에 대한 update가 필요한 것을 예상할 수 있다.

Resample-Move Algorithm은 Sequential Importance Resampling과 Standard Gibbs를 결합하여 다음의 step에 따라 이루어진다.

1) Initialization: t=1 시점에 대한 초기 n개의 독립적인 Sample Set $S_1 = (\tau^{-1}, x_1, y_1, \dot{x}_1, \dot{y}_1)$을 얻는다.
2) Rejuvenation: n개의 sample에 대한 Importance weights (~likelihood)을 계산한다. 이를 이용하여 t=2, 3,⋯ 시점에 대해 다음의 두가지 step을 수행한다.
   (Resample step) t 시점에 계산된 w_t을 이용하여 n개의 sample particle들을 Resampling한다.
   $$w_t = N(tan^{-1}(x_t/y_t), \eta^2) \text{ (likelihood)}$$
   (Move step) parameter들에 대한 stationary distribution을 이용해 각 particle들에 대한 parameter τ를 $x_{1:t}, y_{1:t}$ 에 대한 조건부 Gibbs sampling을 통해 update한다.

시점 t의 target posterior distribution은 다음과 같이 prior와 likelihood의 곱에 비례하며, prior $p(\theta_t)$는 선박의 속도에 대한 분포에 대한 정보를 포함하여 이루어진다.

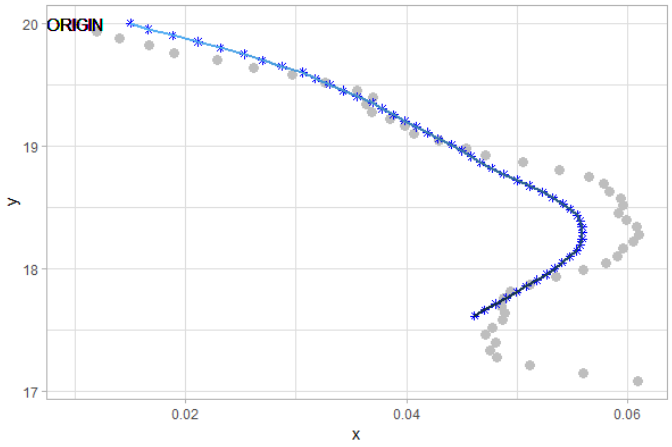
$$\pi_t(\theta_t) = p(\theta\,|z_{1:t}) \propto p(\theta_t) \cdot \prod_{i=1}^{t} p(z_i|\theta_i), \qquad p(\theta_t) \propto \exp\{-0.5\tau\sum(\dot{x}_t - \dot{x}_{t-1})^2 - 0.5\tau\sum(\dot{y}_t - \dot{y}_{t-1})^2\}$$

[1] W. R. Gilks. and C. Berzuini. Following a Moving Target - Monte Carlo inference for dynamic Bayesian models. *Journal of Pattern Recognition and Artifitial intelligence*, 15:9-42, 2001.

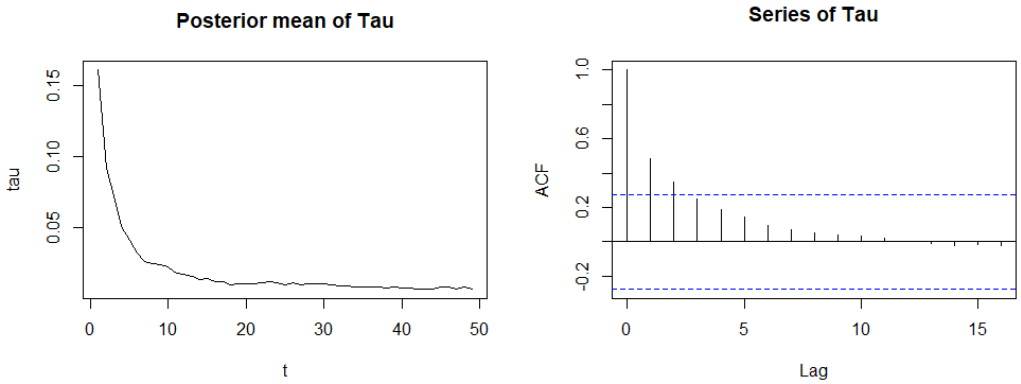
posterior는 t-1시점의 값에만 영향을 받는 one-step transition probability들로 이루어진 stationary distribution이다. 따라서 τ 의 조건부 sampling 시 posterior에 비례한 $\mathrm{Gamma}(2, -0.5\{\sum(\dot{x}_t - \dot{x}_{t-1})^2 - 0.5\tau\sum(\dot{y}_t - \dot{y}_{t-1})^2\})$ 분포를 이용하였다.

실제 시뮬레이션 데이터로서 t=50 시점까지의 관측치를 생성할 때 이용한 수치는 다음과 같다. sampling의 초기값으로 이 실제 값에 대해 약간씩 값을 수정하여 이용하였다.

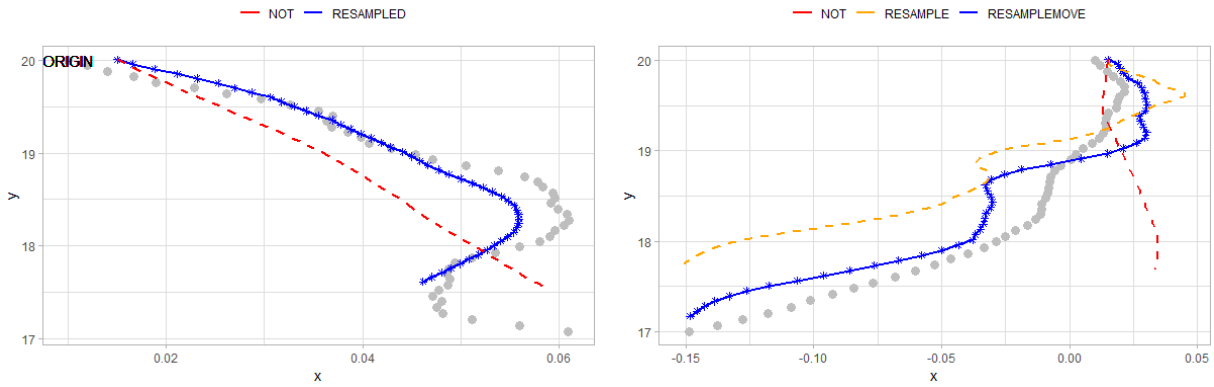
x_1	y_1	$\dot{x}_1$	$\dot{y}_1$	τ^{-1}	η
0.01	20	0.002	-0.06	0.000001	0.005



위에서 설명한 알고리즘에 따라 각 시점에 대해 n=1000개의 sample sets를 생성하였으며 이들의 평균값을 이용해 위치를 추정한 결과, 실제 궤도와 유사한 위치를 찾아가는 것을 확인할 수 있었다. 또한 아래의 그래프에서 시점마다 resample된 x와 y의 sequence를 이용하여 τ 를 조건부 sampling하였을 때 시점이 증가하여 관측치에 대한 정보가 update됨에 따라 실제 값인 0.000001과 가까워지는 것을 확인할 수 있다. 자기상관함수 그래프에서도 correlation이 점점 줄어드는 것을 확인하여 sampling이 효율적으로 진행되었다고 할 수 있다.



Sequential Importance Sampling은 weight에 대한 적절한 회복이 진행되지 않을 경우 추정에 악화를 일으킨다. 따라서 Resample step을 제외하고 기본적인 SIS만을 이용하여 추정하여 결과를 비교해보았다.



그 결과 추정에 매우 차이가 발생하는 것을 확인하였다. 가중치를 계속해서 누적하여 곱하여 평균 계산에 사용한 경우 적절한 추정을 하지 못하였다. 시뮬레이션 데이터를 새롭게 생성하여 각각 SISR(bootstrap filter), Resample and Move, Simple SIS을 적용해 궤도를 추정한 결과 Resample-Move > SISR > SIS 순으로 잘 추정되었다.

3. DISCUSSION

Sequential Importance Sampling은 Random Variable들이 서로 순차적인 sequence를 이룰 때, 이전 시점의 변수를 이용해 다음 시점의 상태를 추정하는 시간적 개념이 포함된 분야의 연구에 활발히 이용된다. 그 대표적인 예시로 금융 및 의학 분야의 시계열 데이터, 움직이는 사물의 궤도에 대한 추정, Control Engineering, Speech Recognition 등 특히 실시간으로 단계적인 예측이 필요한 경우에 유용하게 사용된다. 그러나 앞서 결과에서도 확인하였듯 적절한 타이밍에 Resampling이 진행되지 않을 경우, Importance Weights를 누적하여 사용하는 것은 Simulation에서 생성된 Sample이 일반성을 잃어 추정에 악영향을 미칠 수 있다.

본 프로젝트에서는 Object Tracking Problem 중 하나인 'Bearings-Only Tracking'에 Resampling이 포함된 SISR과 더불어, 다차원의 Target 분포에서도 효과적으로 추정이 가능한 MCMC Simulation의 장점을 결합한 Resample-Move Algorithm을 적용하여 직접 Simulation을 해보았다. 가정한 모형에서 추정하여야 했던 변수들은 50시점까지의 선박의 위치(x-축, y-축 누적속도)와 속도의 일정한 분산을 포함하여 총 101개로 Target Distribution이 상당히 많은 변수를 포함한다. 논문에서 소개된 Resample-Move Algorithm은 다차원의 분포에 대한 실시간적 예측과 함께 알려지지 않은 Hyperparameter에 대해 관측된 데이터를 통해 새롭게 들어오는 정보를 update할 필요가 있는 경우에 효과적으로 이용가능한 방법으로, Sequential IS sampling과 Standard Gibbs Sampling을 일부 결합하여 비교적 계산부담을 줄이면서도 효과적인 추정이 가능하게 하였다.

실제로 Simulation을 구현하면서 Resample step이 진행되지 않을 경우 초기 sample에 크게 영향을 받아 시점이 증가할수록 추정이 점점 더 악화되는 현상이 발생하였다. 또한 Hyperparameter인 속도의 분산을 시점마다 조건부 Sampling으로 Update하여 더 나은 추정 결과를 얻을 수 있었다.

추가적으로 고려하지 못한 한계점으로는 Resampling 및 Hyperparameter τ 의 update가 매 시점 이루어지지 않고 가중치의 Degeneracy를 판단하는 threshold를 도입해본다면 추정치의 분산을 더 줄이고 보다 정확한 추정이 가능하였을 것이라고 예상한다. 또한 시점 t의 증가에 따른 더 많은 parameter들이 모형에 포함될 때 어떤 영향이 있는지도 고려해보아야 한다.

결과적으로 MCMC와 IS의 결합으로 이루어진 Resample-Move Algorithm은 다른 방법들에 비해 효과적으로 궤도를 추정하였고, 실시간 추정을 하면서도 데이터를 통해 update되는 정보를 이용해 Simulation 도중에 값을 update하는 것 또한 가능하여 효과적인 Algorithm이라는 것을 알게 되었다.

Appendix. R code

```
############################## Functions ##############################
# Calculate IS mean, sd (Unstandardized)
estimate <- function(x, w) {
  w <- w / sum(w)
  ismean <- sum(w*x) / sum(w)
  issd <- sqrt( sum(w*(x-ismean)^2) / (1-sum(w^2)) )
  return(c(mcmean = mean(x), mcsd = sd(x), ismean = ismean, issd = issd))
}

# Generate Real Data
paths <- function(n = 100) {
# hyperparameters
  tau = 0.000001 ; eta = 0.005

# initial values
  x = c(0.01, numeric(n-1)) ; y = c(20, numeric(n-1))
  vx = c(0.002, numeric(n-2)) ; vy = c(-0.06, numeric(n-2))

  for (t in 2:n) {
    vx[t] <- rnorm(1, vx[t-1], sqrt(tau)) ; vy[t] <- rnorm(1, vy[t-1], sqrt(tau))
    x[t] <- x[t-1] + vx[t-1] ; y[t] <- y[t-1] + vy[t-1]
  }
  table <- data.frame(t = 1:n, x, y, vx, vy) %>% mutate(z = atan(y/x) + rnorm(n, 0, eta))
  return(table)
}

realxy <- paths(n = 50)
ggplot(realxy) + theme_light() + theme(legend.position = "") +
  scale_color_continuous() + labs(x = "x", y = "y") +
  geom_path(aes(x, y), color = "lightgray", size = 2) +
  geom_point(aes(x, y), size = 3, shape = 1) +
  geom_point(aes(x[1], y[1]), color = "lightgray", shape = 20, size = 10) +
  geom_text(aes(label = "ORIGIN", x[1], y[1]))

z <- realxy$z
```

```
############################## Tracking ##############################

weight <- function(zt, xt, yt) { dnorm(zt, atan(yt/xt), 0.005) }

init <- c(0.000001, 0, 19.9, 0.001, -0.05)
TRACK1 <- function(t, init, niter=1000, RS = T, MOVE = T) {

# initial values
  tau0 <- init[1] ; x0 <- init[2] ; y0 <- init[3] ; vx0 <- init[4] ; vy0 <- init[5]

  # objects
  vx <- matrix(nrow = t, ncol = niter) ; vy <- vx ; x <- vx ; y <- vx ; weights <- vx
  vx[1,] <- vx0 ; vy[1,] <- vy0 ; x[1,] <- x0 ; y[1,] <- y0 ;

  tau <- matrix(tau0, t, niter)
  weights[1,] <- 1 / niter

  for (ti in 2:t) {

    for (n in 1:niter) {
      vx[ti, n] <- rnorm(1, vx[ti-1, n], sqrt(mean(tau[ti,]))) ; x[ti, n] <- x[ti-1, n] + vx[ti, n]
      vy[ti, n] <- rnorm(1, vy[ti-1, n], sqrt(mean(tau[ti,]))) ; y[ti, n] <- y[ti-1, n] + vy[ti, n]

      weights[ti, n] <- weights[ti-1, n] * weight(z[ti], x[ti, n], y[ti, n])

      if (n%%niter==0) print(paste("t =", ti, "(", round(x[ti, n], 3), ", ", round(y[ti, n], 3), ")"), sep = "")
    }
    weights[ti,] <- weights[ti,] / sum(weights[ti,])

    # 1) resample step
    if (RS) {
      resample <- sample(niter, size = niter, prob = weights[ti,], replace = T)

      vx[1:ti,] <- vx[1:ti, resample] ; vy[1:ti,] <- vy[1:ti, resample]
      x[1:ti,] <- x[1:ti, resample] ; y[1:ti,] <- y[1:ti, resample]

      weights[ti,] <- 1 / niter # reset weights
    }
    # 2) move
    if (MOVE) {
      tau[ti,] <- rgamma(niter, 2, 0.5*(sum(diff(vx[ti,])^2) + sum(diff(vy[ti,])^2))) # ~ Exp(0.5(x^2+y^2))
    }

  }
  output <- list(x = x, y = y, tau = rowMeans(tau), weight = weights[t,])
  return(output)
}

true <- c(0.000001, 0.01, 20, 0.002, -0.06)
init <- c(0.0001, 0.015, 20.005, 0.0, -0.05)
```

```
track1 <- TRACK1(t = 50, init = init, niter = 1000, RS = T, MOVE = T)
output <- data.frame(x = rowMeans(track1$x),
                     y = rowMeans(track1$y))

track2 <- TRACK1(t = 50, init = init, niter = 1000, RS = T, MOVE = F)
output2 <- data.frame(x = rowMeans(track2$x),
                      y = rowMeans(track2$y))

track3 <- TRACK1(t = 50, init = init, niter = 1000, RS = F, MOVE = F)
output3 <- data.frame(x = apply(track3$x, 1, function(x) weighted.mean(x, track3$weight)),
                      y = apply(track3$y, 1, function(x) weighted.mean(x, track3$weight)))

track4 <- TRACK1(t = 50, init = init, niter = 1000, RS = F, MOVE = T)
output4 <- data.frame(x = rowMeans(track4$x),
                      y = rowMeans(track4$y))

ggplot(realxy) + theme_light() + theme(legend.position = "top") +
  labs(x = "x", y = "y") +
  scale_color_manual("", values = c(RESAMPLEMOVE = "blue", RESAMPLE = "orange",
                                    MOVE = "green", NOT = "red")) +
  geom_point(aes(x, y), color = "gray", size = 3) +
  geom_text(aes(label = "ORIGIN", x[1], y[1]), color = "black") +
  geom_point(aes(output$x, output$y), color = "blue", shape = 8) +
  geom_path(aes(output$x, output$y, color = "RESAMPLEMOVE"), size = 1) +
  geom_path(aes(output2$x, output2$y, color = "RESAMPLE"), linetype = "dashed", size = 1) +
  geom_path(aes(output3$x, output3$y, color = "NOT"), linetype = "dashed", size = 1) +
  geom_path(aes(output4$x, output4$y, color = "MOVE"), linetype = "dashed", size = 1)

ggplot(realxy) + theme_light() + theme(legend.position = "") +
    labs(x = "x", y = "y") +
    geom_point(aes(x, y), color = "gray", size = 3) +
    geom_text(aes(label = "ORIGIN", x[1], y[1]), color = "black") +
    geom_point(aes(output$x, output$y), color = "blue", shape = 8) +
    geom_path(aes(output$x, output$y, color = desc(t)), size = 1)

plot(track1$tau[-1], type = "l", xlab = "t", ylab = "tau", main = "Posterior mean of Tau")
acf(track1$tau, main = "Series of Tau")

print(isestimate1 <- estimate(track1$y[50,]. track1$weights))
print(isestimate2 <- estimate(track2$y[50,], track2$weights))
print(isestimate3 <- estimate(track3$y[50,], track3$weights))
```