제16강: HMM(Hidden Markov Model)

금융 통계 및 시계열 분석

TRADE INFORMATIX

2014년 3월 4일

목차

독립적 혼합 모형 (Independent Mixture Model)

- 복수의 연속확률분포 중 하나를 확률적으로 선택하는 조합 방법
- 분포 선택은 독립적인 이산 확률분포를 사용

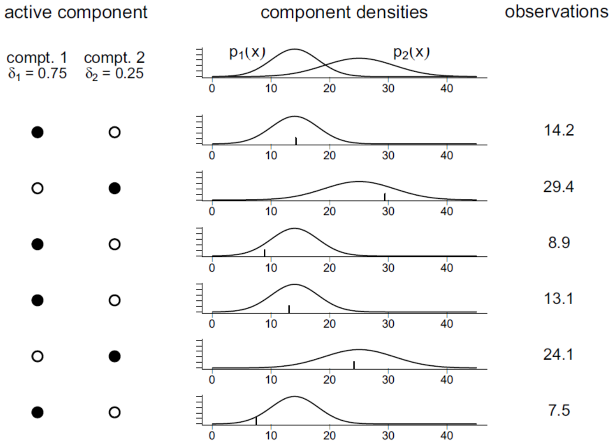
$$p(x) = \text{Pr}(X = x) = \sum_{i=1}^{m} \text{Pr}(C = i) \cdot \text{Pr}(X = x|C = i) = \sum_{i=1}^{m} \delta_i p_i(x)$$

- $p(x)$: 전체 Independent Mixture 분포
- $p_i(x)$: Independent Mixture의 각 성분 (component) 이 되는 개별적인 연속 확률분포
- δ_i : mixing parameter. 특정시간에 대해 모든 성분 중 특정한 $p_i(x)$ 가 선택될 확률
- $\sum \delta_i = 1$: mixing parameter에 대한 확률 제한 조건

독립적 혼합 모형의 예

❏ Binomial Normal-Mixuture 모형

- ► 각 성분은 Normal 연속 분포이고
- ► 성분을 선택하는 이산분포는 Binomial 분포



Mixture 모형의 파라미터 추정

- Maximum Likelihood Estimation 사용
- n 개의 샘플에 대한 Mixture 모형의 Likelihood 함수

$$L(\theta_1, \cdots, \theta_m, \delta_1, \cdots, \delta_m | x_1, \cdots, x_n) = \prod_{j=1}^{n} \sum_{i=1}^{m} \delta_i p_i(x_j, \theta)$$

- 덧셈과 곱셈이 혼합되어 있는 복잡한 함수
- 일반적인 Log-Differentiation 방법 사용 불가
- miximg parameter 제한 조건 준수
- Constrained Non-linear Optimization 또는 EM 알고리즘 사용
- Zero-Variance Spike 문제

마코프 체인 (Markov Chain)

- ❑ 다음과 같은 마코프 (Markov) 특성을 가지는 이산시간 확률 프로세스

$$\text{Pr}(C_{t+1}|C_t, \cdots, C_1) = \text{Pr}(C_{t+1}|C_t)$$

- ❑ 전이확률 (Transition Probability)
  - ▶ 특정 시간 t 동안 특정한 한 상태 i 에서 특정한 다른 상태 j 로 전이할 확률

$$\gamma_{ij}(t) = \text{Pr}(C_{s+t} = j|C_s = i)$$

- ❑ 전이확률행렬 (Transition Probability Matirx)

$$\Gamma(t) = \{\gamma_{ij}(t)\} \quad \Gamma = \Gamma(1)$$

- ❑ Chapman-Kolmogorov Equation
  - ▶ 시간 $t+u$ 의 전이확률행렬은 시간 t 의 전이확률행렬과 시간 u 의 전이확률행렬의 곱

$$\Gamma(t+u) = \Gamma(t)\Gamma(u)$$

마코프 체인 (Markov Chain) (계속)

- Unconditional Probability
  - t 라는 시점에 특정한 이산 상태에 있을 확률, 즉 특정한 component가 선택될 확률

$$u(t) = (\text{Pr}(C_t = 1), \text{Pr}(C_t = 2), \cdots, \text{Pr}(C_t = m))$$

$$u(t+1) = u(t)\Gamma$$

- Stationary Distribution
  - 시간이 지나도 uncoditional probability가 변하지 않는 경우

$$\delta = \delta\Gamma$$

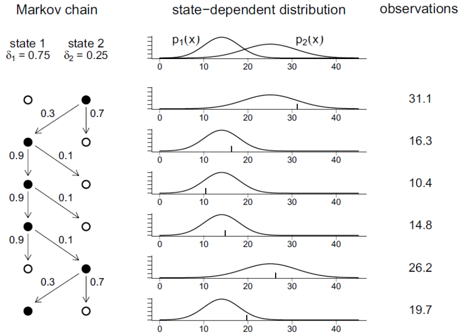
히든 마코프 모형 (Hidden Markov Model)

❑ Hidden Markov Model

- ▶ 연속 확률 분포를 선택하는 파라미터 프로세스 (Parameter Process) 가 마코프 체인 (Markov Chain) 이고 연속확률분포가 그 시점의 파라미터 프로세스의 샘플값에만 의존하는 Mixture Model
- ▶ 연속확률분포의 샘플값만 측정가능

$$\Pr(C_t|C_{t-1},\cdots,C_1) = \Pr(C_t|C_{t-1})$$

$$\Pr(X_t|X_t,\cdots,X_1,C_t,\cdots,C_1) = \Pr(X_t|C_t)$$

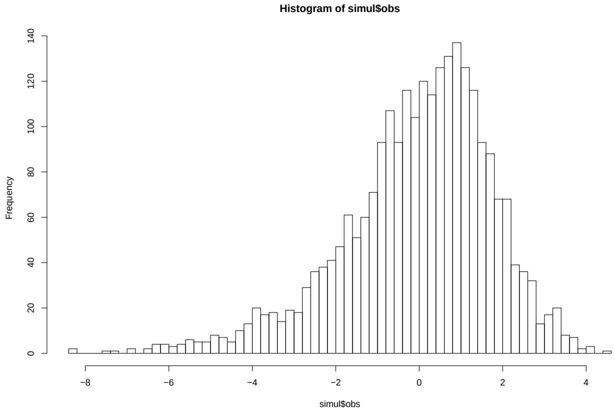


히든 마코프 모형 시뮬레이션

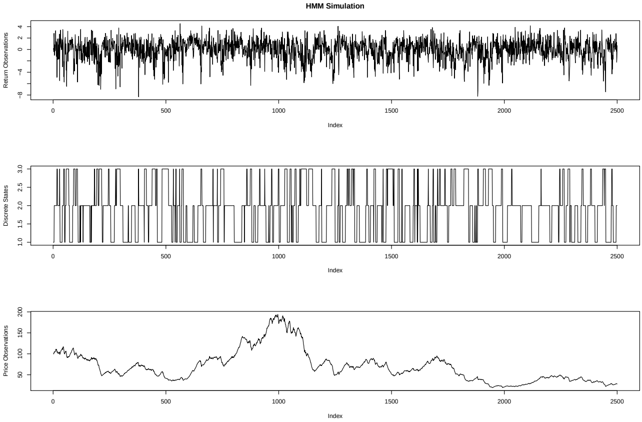
- RHmm 패키지 사용
- `distributionSet` 명령으로 연속분포 집합 생성
- `HMMSet` 명령으로 HMM 오브젝트 생성
- `HMMSim` 명령으로 시뮬레이션
- `distributionSet(dis="NORMAL", mean, variance)`
  - `dis`: 연속분포 모형 `'NORMAL'`, `'DISCRETE'`, `'MIXTURE'`
  - `mean`, `variance`: `dis='NORMAL'`인 경우 파라미터
- `HMMSet(initProb, transMat, distribution)`
  - `initProb`: 초기 이산확률분포
  - `transMat`: 확률전이행렬 (Probability Transition Matrix)
  - `distribution`: `distributionSet` 명령으로 생성한 연속분포 집합
- `HMMSim(nSim, HMM)`
  - `nSim`: 시뮬레이션 횟수
  - `HMM`: `HMMSet` 명령으로 생성한 HMM 오브젝트

히든 마코프 모형 시뮬레이션 예

```
> library("RHmm")
> set.seed(2)
> n_1d_3s <- distributionSet("NORMAL", mean=c(1, 0.05, -2), var=c(1, 2, 4))
> initProb3 <- rep(1,3)/3
> transMat3 <- rbind(c(0.9, 0.09, 0.01), c(0.05, 0.9, 0.05), c(0.05, 0.1, 0.85))
> hmm_1d_3s <- HMMSet(initProb3, transMat3, n_1d_3s)
> simul <- HMMSim(2500, hmm_1d_3s)
> hist(simul$obs, breaks=50)
```



```
> par(mfrow=c(3,1))
> plot(simul$obs, type='l', main="HMM Simulation", ylab="Return Observations")
> plot(simul$states, type='l', ylab="Discrete States")
> plot(100*exp(cumsum(simul$obs/100.0)), type='l', ylab="Price Observations")
```



HMM 파라미터 추정 문제

- 관측된 n 개의 확률변수 샘플 $\{X_i\}, (i = 1, \cdots, n)$ 로부터 component 연속확률분포의 파라미터와 Mixing Paramter Process인 Markov Chain의 파라미터를 동시에 추정하는 문제
- Mixing Paramter Process는 Probability Transition Matrix의 형태로 추정
- Probability Transition Matrix는 m^2 의 파라미터를 가지지만 실제로는 각 행의 합이 1 이므로 $m(m-1)$ 의 파라미터가 된다.

$$L(\theta_1, \cdots, \theta_m, \gamma_{11}, \cdots, \gamma_{mm} | x_1, \cdots, x_n)$$

- Maximum Likelihood Estimation 방법론 이용
- 다양한 제한조건이 존재
- EM Mothod (Baum-Welch Method) 사용

HMM Likelihood

❑ 샘플 $\{x_1, x_2, \cdots, x_T\}$가 주어졌을 때 이 샘플에 대한 Likelihood 함수 L_T는 다음과 같이 주어진다.

$$\begin{aligned} L_T &= u(1)P(x_1)\Gamma P(x_2)\Gamma P(x_3)\cdots\Gamma P(x_T)1^T \\ &= u(1)P(x_1)\prod_{s=2}^{m}\Gamma P(x_s)1^T \end{aligned}$$

- $u(1)$: $t = 1$에서의 component 선택 확률

$$u(1) = (\mathrm{Pr}(C_1 = 1), \mathrm{Pr}(C_1 = 2), \cdots, \mathrm{Pr}(C_1 = m))$$

- $P(x)$: 각각의 component분포가 x이 될 확률을 대각성분으로 가지는 대각행렬

$$P(x) = \mathrm{diag}\left(p_1(x), p_2(x), \cdots, p_m(x)\right)$$

HMM MLE의 어려움

- 제한 조건 문제
  - Probability Transition Matrix의 모든 원소는 0과 같거나 크다

$$\gamma_{ij} \geq 0 \ \text{ for all } i, j$$

  - Probability Transition Matrix의 각 행의 값의 합은 1

$$\sum_{i=1}^{m} \gamma_{ij} = 1$$

  - 제한 조건 문제를 풀기 위해 다음과 같은 변환 사용

$$\begin{aligned}
\gamma_{ij} &= \frac{\rho_{ij}}{\sum_{k=1}^{m} \rho_{ik}} \\
\rho_{ij} &= \begin{cases} e^{\tau_{ij}} & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \\
\tau_{ii} &= 0
\end{aligned}$$

  - $\{\tau_{ij}\}$ 행렬을 추정한 뒤 $\Gamma = \{\gamma_{ij}\}$ 행렬로 변환

EM Algorithm

- 관측할 수 없는 discrete state를 추정한 후 이를 이용하여 다시 모형 파라미터를 추정하는 것을 반복
  - E-step: discrete state $C^{(T)}$ 를 추정

$$\hat{u}(t) = \mathrm{Pr}(C_t = j | X^{(T)})$$

  - M-step: 추정된 $u(t)$ 가 맞다는 가정하에 MLE 방법으로 파라미터 추정

$$\arg\max_{\theta} L_T(\theta | X^{(T)}, C^{(T)})$$

HMM 추정 명령

- RHmm 패키지 사용
- `HMMFit(obs, dis="NORMAL", nStates=2)`
  - `obs`: 관측값
  - `dis`: 연속분포 모형 `'NORMAL'`, `'DISCRETE'`, `'MIXTURE'`
  - `nStates`: 이산분포 state 수
- 출력
  - `HMM`: HMM 파라미터
    - `initProb`: 초기확률분포
    - `transMat`: 전이확률행렬
    - `distribution`: 연속확률분포 집합의 파라미터

      `mean`: mean

      `var`: variance

HMM 추정 예

```
> (m <- HMMFit(simul$obs, "NORMAL", 3))

Call:
----
HMMFit(obs = simul$obs, dis = "NORMAL", nStates = 3)

Model:
------
3 states HMM with univariate gaussian distribution

Baum-Welch algorithm status:
----------------------------
Number of iterations : 85
Last relative variation of LLH function: 0.000001

Estimation:
-----------

Initial probabilities:
          Pi 1        Pi 2 Pi 3
  1.277975e-87 3.06182e-22    1

Transition matrix:
            State 1    State 2    State 3
State 1 0.801038237 0.09657887 0.10238289
State 2 0.059976843 0.90135999 0.03866316
State 3 0.006631079 0.08464813 0.90872079

Conditionnal distribution parameters:

Distribution parameters:
              mean      var
State 1 -2.3373954 4.274560
State 2 -0.1285315 2.087011
State 3  0.9847118 1.086262

Log-likelihood: -4684.26
BIC criterium: 9478.05
AIC criterium: 9396.52
```

Decoding

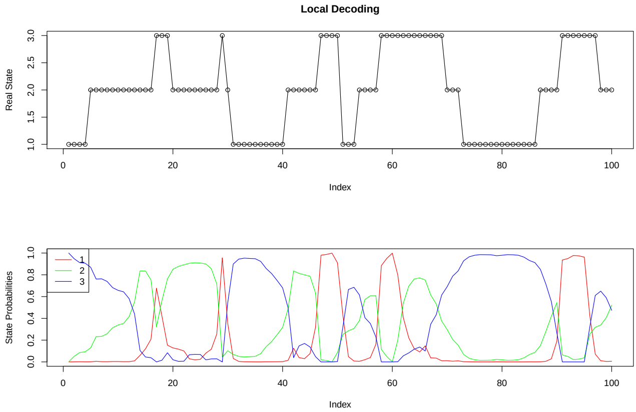
- 추정된 HMM 파라미터를 이용하여 마코프 체인의 discrete state를 추정하는 것을 **디코딩 (Decoding)**이라 한다.
  - Local Decoding
    - 특정 시간 t에서 각 discrete state가 될 확률을 구하는 문제
  - Global Decoding (Path Finding)
    - 가장 Likelihood가 높은 마코프 체인의 discrete state 경로를 재현하는 문제
    - 비터비 (Viterbi) 알고리즘 사용

Local Decoding 명령

- RHmm 패키지 사용
- `forwardBackward(HMM, obs)`
  - `HMM`: `HMMFit` 명령으로 생성한 HMM 오브젝트
  - `obs`: 관측값
- 출력
  - `Alpha`: α
  - `Beta`: β
  - `LLH`: Log-Likelihood

Local Decoding 예

```
> fb <- forwardBackward(m, simul$obs)
> local <- exp(fb$Alpha+fb$Beta-fb$LLH)
> par(mfrow=c(2,1))
> plot(simul$states[1:100], type='o', main="Local Decoding", ylab="Real State")
> plot(local[1:100,1], type='l', col="red", ylab="State Probabilities")
> lines(local[1:100, 2], col="green")
> lines(local[1:100, 3], col="blue")
> legend("topleft", c("1", "2", "3"), col=c("red", "green", "blue"), lty=1)
```

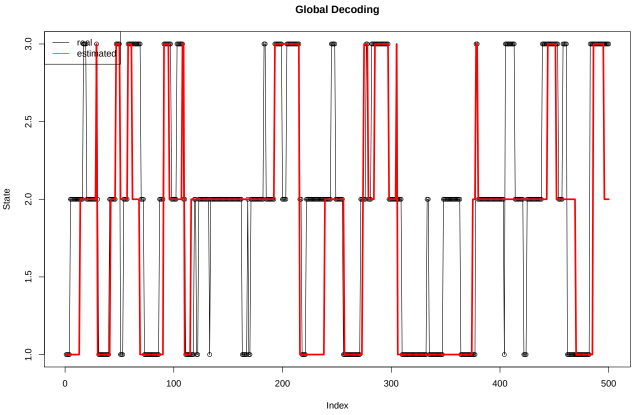


Global Decoding 명령

- RHmm 패키지 사용
- `viterbi(HMM, obs)`
  - `HMM`: `HMMFit` 명령으로 생성한 HMM 오브젝트
  - `obs`: 관측값
- 출력
  - `states`: state path

Global Decoding 예

```
> vitpath <- viterbi(m, simul$obs)
> states <- vitpath$states
> states[states==1] <- 0; states[states==3] <- 1; states[states==0] <- 3
> plot(simul$states[1:500], type='o', main="Global Decoding", ylab="State")
> lines(states[1:500], col="red", lwd=3, pch=4)
> legend("topleft", c("real", "estimated"), col=c("black", "red"), lty=1)
```

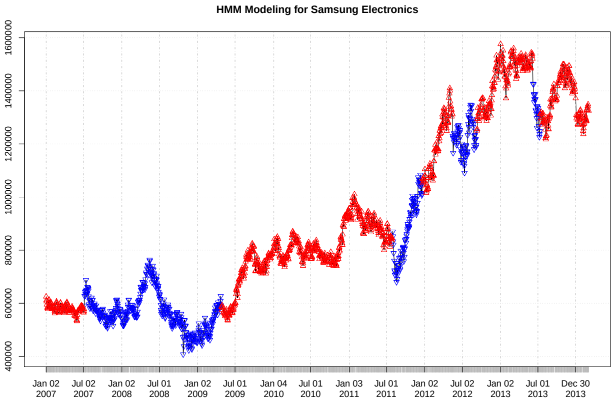


실제 주가에 HMM 응용

```
> library(quantmod)
> library(TTR)
> x <- getSymbols("KRX:005930", src="google", auto.assign=FALSE)
> p <- x$"KRX:005930.Close"
> r <- na.fill(coredata(ROC(p)),0)
> (sec.m <- HMMFit(r, "NORMAL", 2))

Call:
----
HMMFit(obs = r, dis = "NORMAL", nStates = 2)

Model:
------
2 states HMM with univariate gaussian distribution

Baum-Welch algorithm status:
----------------------------
Number of iterations : 69
Last relative variation of LLH function: 0.000001

Estimation:
-----------

Initial probabilities:
          Pi 1 Pi 2
  9.911473e-35    1

Transition matrix:
           State 1   State 2
State 1 0.97810920 0.0218908
State 2 0.01141105 0.9885890

Conditionnal distribution parameters:

Distribution parameters:
                 mean          var
State 1 -0.0002057263 0.0007788063
State 2  0.0007477859 0.0002484962

Log-likelihood: 4469.61
BIC criterium: -8886.81
AIC criterium: -8925.21
```

실제 주가에 HMM 응용

```
> sec.path <- viterbi(sec.m, r)
> plot(p, main="HMM Modeling for Samsung Electronics")
> points(p[sec.path$states == 1], pch=6, col="blue")
> points(p[sec.path$states == 2], pch=2, col="red")
```



실제 주가에 HMM 응용

```
> library(quantmod)
> library(TTR)
> x <- getSymbols("INDEXKRX:KOSPI200", src="google", auto.assign=FALSE)
> p <- x$"INDEXKRX:KOSPI200.Close"
> r <- na.fill(coredata(ROC(p)),0)
> (sec.m <- HMMFit(r, "NORMAL", 2))

Call:
----
HMMFit(obs = r, dis = "NORMAL", nStates = 2)

Model:
------
2 states HMM with univariate gaussian distribution

Baum-Welch algorithm status:
----------------------------
Number of iterations : 10
Last relative variation of LLH function: 0.000000

Estimation:
-----------

Initial probabilities:
          Pi 1 Pi 2
  3.712691e-17    1

Transition matrix:
           State 1    State 2
State 1 0.97638991 0.02361009
State 2 0.00668076 0.99331924

Conditionnal distribution parameters:

Distribution parameters:
                mean          var
State 1 -0.001465504 0.0006846718
State 2  0.000645993 0.0001092940

Log-likelihood: 5188.82
BIC criterium: -10325.24
AIC criterium: -10363.64
```

실제 주가에 HMM 응용

```
> sec.path <- viterbi(sec.m, r)
> plot(p, main="HMM Modeling for KOSPI200")
> points(p[sec.path$states == 1], pch=6, col="blue")
> points(p[sec.path$states == 2], pch=2, col="red")
```