

```

```{r echo=F, eval=F}
A=matrix(c(0.5, 0.2,0.3,0.3,0.5,0.2,0.2,0.3,0.5), byrow = T, nrow=3)
B=matrix(c(0.5,0.5,0.4,0.6,0.7,0.3),byrow=T, nrow=3)
pi=c(0.2,0.4,0.4)
assume B1 B2 B1

forward algorithm example
gamma1 <- pi * B[,1]

phi21=which.max(gamma1 * A[,1]) # t=2 的一来自 t=1 的三
gamma21=(gamma1 * A[,1])[phi21] * B[1,2] # 前一个用最大可能变成 1 的概率 * 下一个 1 抽 B2 的概率
phi22=which.max(gamma1 * A[,2]) # t=2 的二来自 t=1 的三
gamma22=(gamma1 * A[,2])[phi22] * B[2,2] # 前一个用最大可能变成 2 的概率 * 下一个 2 抽 B2 的概率
phi23=which.max(gamma1 * A[,3]) # t=2 的三来自 t=1 的三
gamma23=(gamma1 * A[,3])[phi23] * B[3,2]

phi31=which.max(gamma2 * A[,1]) # t=3 的一来自 t=2 的二
gamma31=(gamma2 * A[,1])[phi31] * B[1,1] # 前一个用最大可能变成 1 的概率 * 下一个 1 抽 B2 的概率
phi32=which.max(gamma2 * A[,2]) # t=3 的二来自 t=2 的二
gamma32=(gamma2 * A[,2])[phi32] * B[2,1] # 前一个用最大可能变成 2 的概率 * 下一个 2 抽 B2 的概率
phi33=which.max(gamma2 * A[,3]) # t=3 的三来自 t=2 的三
gamma33=(gamma2 * A[,3])[phi33] * B[3,1]

which.max(c(gamma31, gamma32, gamma33)) # t=3 时三最有可能性,往上追溯得到最佳路径

b=c(1,2,1)
phi=matrix(nrow=3,ncol=2)
gamma <- matrix(nrow=3, ncol=3)
gamma[,1] <- pi * B[,b[1]]

for (i in 1:2) {
 phi[,i] <- apply(gamma[,i]*A, 2, which.max)
 gamma[,i+1] <- apply(gamma[,i]*A, 2, max) * B[,b[i+1]]
}

vpath <- matrix(ncol=3)
vpath[3] <- which.max(gamma[,3])
for(i in 2:1){
 vpath[i] <- phi[vpath[i+1],i]
}
```

```