FE 570 Group Project: PIN Model

Teegala Mahender, Niresh, Harshill, Divyansh

Introduction

A theoretical model called PIN (Probability of Informed Trading) aims to calculate the likelihood that a transaction is conducted by someone who has access to information that is not generally known. In the financial markets, this approach is used to assist spot possible insider trading and market manipulation.

The PIN model is predicated on the idea that traders who are knowledgeable have an advantage over those who are not and are thus more likely to make lucrative deals. The model calculates the likelihood that a deal was executed by a knowledgeable trader using publicly accessible data, such as trading volume and price fluctuations.

Parameters for Pin Model

a. α — news arrival probability

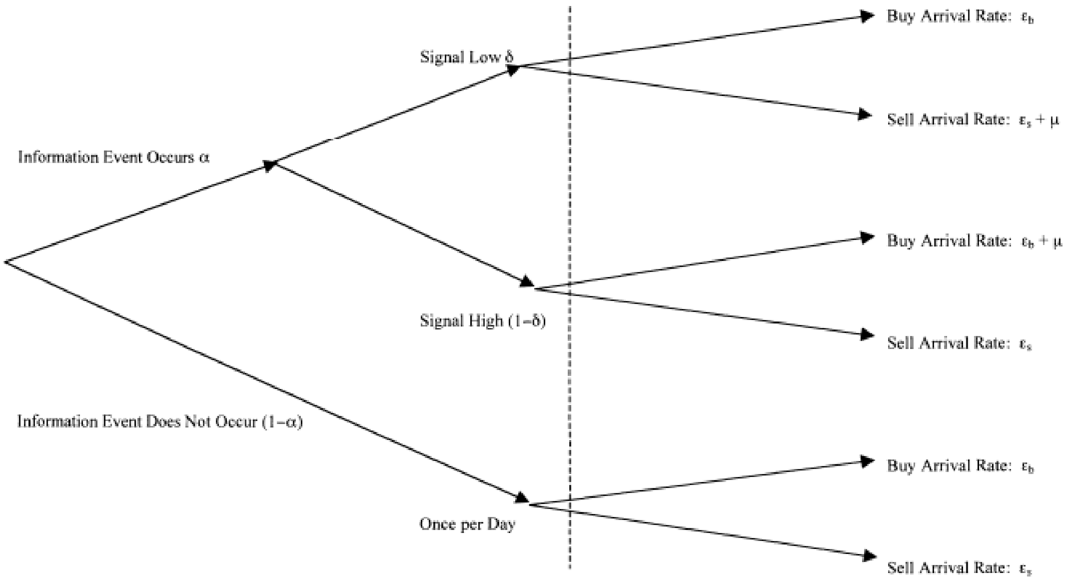
b. δ — the probability that the news is a low signal

c. μ — the arrival rate of informed traders

d. εb — the arrival rate of uninformed traders that buy

e. εs — the arrival rate of uninformed traders that sell

Event Tree for Model



Loading Required Libraries

```
---
title: "Project"
author: "Teegala Mahender"
date: "2023-05-08"
---




```{r}

library(highfrequency)
library(InfoTrad)


```



```{r}

library(data.table)
library(xts)
library(TTR)
library(timeDate)
library(quantmod)

```



```

## Estimating Parameters for Pin

The combined probability distribution in relation to the parameter vector  and the $\Theta \equiv \{\alpha, \delta, \mu, \epsilon_b, \epsilon_s\}$ number of $B_t$ and $S_t$ buys and sells is specified by

$$f(B_t, S_t|\Theta) \equiv \alpha\delta exp(-\epsilon_b)\frac{\epsilon_b^{B_t}}{B_t!}exp[-(\epsilon_s+\mu)]\frac{(\epsilon_s+\mu)^{S_t}}{S_t!}$$
$$+\alpha(1-\delta)exp[-(\epsilon_b+\mu)]\frac{(\epsilon_b+\mu)^{B_t}}{B_t!}exp(-\epsilon_s)\frac{\epsilon_s^{S_t}}{S_t!}$$
$$+(1-\alpha)exp(-\epsilon_b)\frac{\epsilon_b^{B_t}}{B_t!}exp(-\epsilon_s)\frac{\epsilon_s^{S_t}}{S_t!}$$

| Parameter_List | simulation_constant_mu1 | Est_100 | simulation_constant_mu2 | Est_200 | simulation_constant_mu3 | Est_300 |
|---|---|---|---|---|---|---|
| Alpha | 0.34 | 0.310002 | 0.34 | 0.4 | 0.34 | 0.330001843 |
| Delta | 0.66 | 0.774191 | 0.66 | 0.575002 | 0.66 | 0.727271292 |
| Epsilon_buy | 100 | 99.14995 | 100 | 100.3338 | 100 | 99.89472105 |
| Epsilon_sell | 100 | 100.6325 | 100 | 98.80568 | 100 | 101.138957 |
| Mu | 100 | 97.86297 | 200 | 197.9265 | 300 | 299.4737569 |
| PIN | 0.145299145 | 0.131834 | 0.253731343 | 0.284469 | 0.337748344 | 0.329576149 |

- The likelihood that a good or poor news event will occur, as well as the future arrival rates, are described in the event tree. In this model, represents the chance of the news arriving, is the likelihood that the news is a weak signal, is the arrival rate of knowledgeable traders, and b

- is the arrival rate of uneducated buyers, and s is the arrival rate of uneducated sellers. In order to replicate this, we wrote a function that generates series of 100 transactions up to 300 trades while maintaining a constant value for all other parameters. Next, we utilized the EA function in the InfoTrad package to estimate these parameters.

```
library(highfrequency)
library(data.table)

library(xts)

library(TTR)
library(timeDate)

library(quantmod)

library(InfoTrad)
```

```
##
## Attaching package: 'InfoTrad'

## The following object is masked from 'package:base':
##
##     print

simulation_pin <- function(n, alpha, delta, epsilon_b, epsilon_s, mu) {
  set.seed(123) # add a seed for reproducibility
  rand1 <- runif(n)
  news <- rand1 <= alpha
  no_news <- rand1 > alpha
  rand2 <- runif(sum(news))
  lo_news <- rand2 <= delta
  hi_news <- rand2 > delta

  no_of_buys <- rep(0, n)
  no_of_sells <- rep(0, n)
  lo_buy <- rep(0, n)
  lo_sell <- rep(0, n)
  hi_buy <- rep(0, n)
  hi_sell <- rep(0, n)

  lo_buy[news][lo_news] <- rpois(sum(lo_news), epsilon_b)
  lo_sell[news][lo_news] <- rpois(sum(lo_news), epsilon_s + mu)
  hi_buy[news][hi_news] <- rpois(sum(hi_news), epsilon_b + mu)
  hi_sell[news][hi_news] <- rpois(sum(hi_news), epsilon_s)
  no_of_buys[no_news] <- rpois(sum(no_news), epsilon_b)
  no_of_sells[no_news] <- rpois(sum(no_news), epsilon_s)

  buys <- no_of_buys + lo_buy + hi_buy
  sell <- no_of_sells + lo_sell + hi_sell

  return(cbind(buys, sell))
}

n <- 100
alpha <- 0.34
delta <- 0.66
epsilon_b <- 100
epsilon_s <- 100
mu <- 100

data_100 <- simulation_pin(n, alpha, delta, epsilon_b, epsilon_s, mu)
data_200 <- simulation_pin(n, alpha, delta, epsilon_b, epsilon_s, mu=200)
data_300 <- simulation_pin(n, alpha, delta, epsilon_b, epsilon_s, mu=300)

pin_100 <- alpha * 100 / (alpha * 100 + epsilon_b + epsilon_s)
pin_200 <- alpha * 200 / (alpha * 200 + epsilon_b + epsilon_s)
pin_300 <- alpha * 300 / (alpha * 300 + epsilon_b + epsilon_s)

res_100 <- EA(data_100)
res_200 <- EA(data_200)
res_300 <- EA(data_300)
```

```
res_100$PIN <- (res_100$alpha * res_100$mu) / (res_100$alpha * res_100$mu
+ res_100$epsilon_b + res_100$epsilon_s)
res_200$PIN <- (res_200$alpha * res_200$mu) / (res_200$alpha * res_200$mu
+ res_200$epsilon_b + res_200$epsilon_s)
res_300$PIN <- (res_300$alpha * res_300$mu) / (res_300$alpha * res_300$mu
+ res_300$epsilon_b + res_300$epsilon_s)

Parameter_List <- c("Alpha", "Delta", "Epsilon_buy", "Epsilon_sell", "Mu",
"PIN")
simulation_constant_mu1 <- c(alpha, delta, epsilon_b, epsilon_s, 100, pin_
100)
```

Volatility

- We have estimated the volatility using Roll model

- ```
  #######volatility

  p <- as.numeric(tqdata$PRICE)

  dp = diff(p)

  # compute the covariance of the price changes, for the Roll model an
  alysis
  covdp <- acf(dp, lag.max=10,
               type="covariance", plot=FALSE,
               main="Autocovariance of price changes")

  gamma0 <- covdp$acf[1]
  gamma1 <- covdp$acf[2]
  n.trades <- dim(tqdata)[1]
  sig2u = gamma0 + 2*gamma1

  rvRoll <- sig2u*n.trades
  print(rvRoll)
  ```
- ```
  ## [1] 0.4940352
  ```

We estimated the intraday volatility of the stock to be 0.49.

Liquidity measures:

Liquidity measures are used to estimate how well the stock is being traded in the market.

```r
liq.m <- getLiquidityMeasures(tqdata)
```

```r
# compute the average Effective Spread
average.es <- mean(as.numeric(liq.m$effectiveSpread))
average.es

## [1] 0.007887397
```

We have established the effective spread of the stock for the stock on 03/01/2023 to be 0.00788.

Pin Estimation:

Formula for pin estimation:

$$\text{PIN}_t = \frac{\alpha\mu}{\alpha\mu + 2\epsilon}$$

```r
#   load data set
options(digits.secs=3)

#load("C:/Users/Mahender/Downloads/KO03012022_trade_quote (2).csv")

# loads a file called tqdata

Sys.setenv(TZ='GMT') # added to remove warnings about time zone mismatch

data<-read.csv("C:/Users/Mahender/Downloads/KO03012022_trade_quote
(2).csv") # Read the dataset
data<-as.data.table(data)
# Change the Column names according to the TAQ format
colnames(data)[1]<-"SYMBOL"
colnames(data)[3]<-"DT"
colnames(data)[7]<-"PRICE"
colnames(data)[8]<-"SIZE"
colnames(data)[10]<-"BID"
colnames(data)[11]<-"BIDSIZE"
colnames(data)[13]<-"OFR"
colnames(data)[14]<-"OFRSIZE"

# Convert the Date Time Column to time-series format
trades<-as.POSIXct(data$DT,format="%Y-%m-%dT%H:%M:%OSZ",tz="GMT")- 5*3600

data$DT=trades
head(data)
```

```r
# Subset the Trades table from the given dataset
tdata<-subset(data, Type=="Trade")
#Remove the unnecessary columns
tdata<-tdata[,-c(2,4,5,6,9:14)]

tdata<-unique(tdata)
head(tdata)

# Subset the Quotes table from the given dataset
qdata<-subset(data, Type=="Quote")
#Remove the unnecessary columns
qdata<- qdata[,-c(2,4,5,6,7,8,9,12)]

qdata$MIDQUOTE<-(as.numeric(qdata$BID)+as.numeric(qdata$OFR))/2
qdata<- unique(qdata)
head(qdata)

# Merge trade and quote data
tdata = na.omit(tdata)
qdata = na.omit(qdata)
tqdata<-matchTradesQuotes(tdata, qdata)
head(qdata)

tqdata<-tqdata[-c(1,2,3),]
head(tqdata)

x <- getTradeDirection(tqdata)

tradeDirection <- matrix(x)

buy_side <- which(tradeDirection >0)

num_buy_side <- length(matrix(buy_side))
num_sell_side <- length(tradeDirection) - length(matrix(buy_side))

ntrades <- cbind(num_buy_side, num_sell_side)

ntrades
```

```r
# run optimization of Likelihood function

Buy <- c(350,250,500,552)
Sell <- c(382, 500, 463, 550)
data = cbind(Buy,Sell)


par0 = c(0.5,0.5,300,400,500)

# Call EHO function
EHO_out = EHO(data)
model = optim(par0, EHO_out, gr = NULL,
              method = c("BFGS"), hessian = FALSE)
```

```r
## Parameter Estimates
model$par[1] # Estimate for alpha

model$par[2] # Estimate for delta

model$par[3] # Estimate for mu

model$par[4] # Estimate for eb

model$par[5] # Estimate for es

## Estimate for PIN
(model$par[1]*model$par[3])/((model$par[1]*model$par[3])+model$par[4]+mode
l$par[5])

## [1] 0.1211554
```

Conclusion:

The probability of informed traders on this day was just 12.12%. This suggests that there were more uninformed traders than informed traders.

The pin model requires assumption about the market efficiency and the accuracy of the public information. Improvements of the pin model try to address these limitations.