InfoTrad: An R package for estimating the probability of informed trading

by Duygu Çelik and Murat Tiniç

Abstract The purpose of this paper is to introduce the R package **InfoTrad** for estimating the probability of informed trading (PIN) initially proposed by Easley et al. (1996). PIN is a popular information asymmetry measure that proxies the proportion of informed traders in the market. This study provides a short survey on alternative estimation techniques for the PIN. There are many problems documented in the existing literature in estimating PIN. **InfoTrad** package aims to address two problems. First, the sequential trading structure proposed by Easley et al. (1996) and later extended by Easley et al. (2002) is prone to sample selection bias for stocks with large trading volumes, due to floating point exception. This problem is solved by different factorizations provided by Easley et al. (2010) (EHO factorization) and Lin and Ke (2011) (LK factorization). Second, the estimates are prone to bias due to boundary solutions. A grid-search algorithm (YZ algorithm) is proposed by Yan and Zhang (2012) to overcome the bias introduced due to boundary estimates. In recent years, clustering algorithms have become popular due to their flexibility in quickly handling large data sets. Gan et al. (2015) propose an algorithm (GAN algorithm) to estimate PIN using hierarchical agglomerative clustering which is later extended by Ersan and Alici (2016) (EA algorithm). The package **InfoTrad** offers LK and EHO factorizations given an input matrix and initial parameter vector. In addition, these factorizations can be used to estimate PIN through YZ algorithm, GAN algorithm and EA algorithm.

Introduction

The main aim of this paper is to present the **InfoTrad** package that estimates the probability of informed trading (PIN) initially proposed by Easley et al. (1996). PIN is one of the primary measures of proxy information asymmetry in the market. The structural model is driven from maximum likelihood estimation (MLE). Wide range of studies use PIN to answer questions in different fields of finance[1].

Although it is a heavily used measure in the finance literature, the development of applications that calculate PIN are quite slow. An initial attempt for R community is made by Zagaglia (2012). **FinAsym** package of Zagaglia (2012) and the **PIN** package of Zagaglia (2013) provide the trade classification algorithm of Lee and Ready (1991) which is an important tool for studies that use the TAQ database. Both packages also provide PIN estimates through `pin_likelihood()` functions. However, those estimates are prone to bias due to misspecification and other limitations. **InfoTrad** package aims to overcome such limitations and provide users with a wide range of options when estimating PIN.

Due to the popularity of the measure, problems in estimating PIN recently gained attention in the finance literature. Easley et al. (2010) indicate that for stocks with a large trading volume, it is not possible to estimate PIN due to floating-point-exception (FPE). Two different numerical factorizations are provided by Easley et al. (2010) and Lin and Ke (2011) to overcome the bias created due to FPE.

In addition, boundary solutions in estimating PIN are also shown to create bias in empirical studies. Yan and Zhang (2012) show that, independent of the type of factorization, the likelihood function can stuck at local optimum and provide biased PIN estimates. They propose an algorithm (YZ algorithm) that spans the parameter space by using 125 different initial values for the MLE problem and obtain the PIN estimate that gives the highest likelihood value with non-boundary solutions. Although YZ algorithm provides estimates with higher likelihood and guarantees obtain non-boundary solutions, the iterative structure makes this algorithm time-consuming especially for studies that use large datasets.

Considering the fact that recent studies that estimate PIN use large datasets, the effectiveness of the YZ algorithm is questioned. In recent years, clustering algorithms have become popular due to their efficiency in processing large sets of data. Gan et al. (2015) propose an algorithm that use hierarchical agglomerative clustering to estimate PIN. Ersan and Alici (2016) later extends this framework.

FPE and boundary solutions are *not* the only problems of PIN model. Duarte and Young (2009) indicate that the structural model of Easley et al. (1996) enforces a negative contemporaneous covariance between intraday buy and sell orders, which is contrary to the empirical evidence for symmetric order shocks. In addition,they show that the PIN model fails to capture the volatility of buy and sell orders,through simulations. Moreover, Duarte and Young (2009) adjust PIN to take into account the liquidity impact and show that liquidity is more prominent on stock returns compared to information

[1]For instance, analyst coverage (Easley et al., 1998), stock splits (Easley et al., 2001), initial public offerings (Ellul and Pagano, 2006), credit ratings (Odders-White and Ready, 2006), M&A announcements (Aktas et al., 2007) and asset returns [(Easley et al., 2002),(Easley et al., 2010)] among others.

asymmetry. Finally, it is important to note that PIN does not consider any strategic behaviour of investors such as order splitting. Order splitting can be more evident when a stock is jointly trading on multiple venues (Menkveld, 2008). Even for a stock that is traded on a single market, an informed investor may want to split her order in order avoid revealing her private information too quickly (Foucault et al., 2013). PIN model, by construction, fails to attach multiple small orders to a single informed investor.

This paper introduces and discusses the R (R Core Team, 2016) **InfoTrad** package for estimating PIN. **InfoTrad** provides users with the necessary methods to *solely* adress the problems of FPE and boundary solutions. The package contains the likelihood factorizations of EHO and LK as separate functions (`EHO()` and `LK()`, respectively) which provide likelihood specifications to avoid FPE. In addition, through `YZ()`, `GAN()` and `EA()` functions, PIN estimates can be obtained using the grid-search algorithm of Yan and Zhang (2012) and clustering algorithms of Gan et al. (2015) and Ersan and Alici (2016). For all of the algorithms, likelihood specification can be set to EHO or LK.

The paper is organized as follows; Section 2.2 provides a brief description of PIN. Specifically, section 2.2.1 discusses the problem of FPE and the alternative factorizations EHO and LK. Section 2.2.2 reviews the problem of boundary solutions and the YZ algorithm. Section 2.2.3 describes the clustering algorithms of Gan et al. (2015) and Ersan and Alici (2016). Section 2.3 introduces the package **InfoTrad** along with examples. Section 2.4 evaluates the performance of each method through simulations. Section 2.5 provides concluding remarks.

PIN Model

The structural model of Easley et al. (1996) and Easley et al. (2002) consists of three types of agents; informed traders, uninformed traders and market makers. On a trading day t, one risky asset is continuously traded. Market maker sets the price for a given stock by observing the buy orders (B_t) and sell orders (S_t). For that stock, an information event is assumed to follow a Bernoulli distribution with success probability α. This event reveals either a high or a low signal for the stock value. The event is assumed to provide a low signal with probability δ. When informed traders observe a high (low) signal, they are assumed to place buy (sell) orders at a rate of μ. Uninformed traders are assumed to place orders, independent of the information event and the signal. They arrive to market to place a buy (sell) order at a rate of ϵ_b (ϵ_s). Orders of both informed and uninformed investors are assumed to follow independent Poisson processes.

The joint probability distribution with respect to the parameter vector $\Theta \equiv \{\alpha, \delta, \mu, \epsilon_b, \epsilon_s\}$ and the number of buys and sells (B_t, S_t), is specified by

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

The estimates of arrival rates ($\hat{\mu}, \hat{\epsilon_s}$ and $\hat{\epsilon_b}$), along with estimates of the probabilities ($\hat{\alpha}$ and $\hat{\delta}$) can be obtained by maximizing the joint log-likelihood function given the order input matrix (B_t, S_t) over T trading days. The non-linear objective function of this problem can be written as;

$$L(\Theta|T) \equiv \sum_{t=1}^{T} L(\Theta|(B_t, S_t)) = \sum_{t=1}^{T} log[f(B_t, S_t|\Theta)] \tag{2}$$

The maximization problem is subject to the boundary constraints $\alpha, \delta \in [0,1]$ and $\mu, \epsilon_b, \epsilon_s \in [0, \infty)$[2]. The PIN estimate is then given by;

$$\widehat{PIN} = \frac{\hat{\alpha}\hat{\mu}}{\hat{\alpha}\hat{\mu} + \hat{\epsilon_b} + \hat{\epsilon_s}} \tag{3}$$

[2]Both **PIN** package of Zagaglia (2013) and **FinAsym** package of Zagaglia (2012) fail to acknowledge the boundary constraints on arrival rates $\mu, \epsilon_b, \epsilon_s$. Similar to event probabilities, they restrict these parameters to $[0,1]$ which forces the estimates for the arrival of informed and uninformed traders on a given day to take values at most one. This creates significant bias in PIN estimates.

Floating-Point Exception

PIN estimates are prone to selection bias, especially for stocks for which the number of buy and sell orders are large[3]. Lin and Ke (2011) show that the increase in the number of buy and sell orders for a given stock, significantly shrinks the feasible solution set for the maximization of the log likelihood function in equation (2). To maximize the non-linear function (1), the optimization software introduces initial values for the parameters in Θ. The numerical optimization method is applied after those initial parameters are introduced. Therefore, for large enough B_t and S_t whose factorials cannot be calculated by mainstream computers (i.e. FPE), the optimal value for equation (2) becomes undefined. The FPE problem is therefore, more pronounced in active stocks.

To avoid the bias created due to FPE, one factorization of the equation (2) is provided by Easley et al. (2010) as $L_{EHO}(\Theta|T) \equiv \sum_{t=1}^{T} L_{EHO}(\Theta|B_t, S_t)$ where

$$\begin{aligned} L_{EHO}(\Theta|B_t, S_t) = log[\alpha\delta exp(-\mu)x_b^{B_t-M_t}x_s^{-M_t} + \alpha(1-\delta)exp(-\mu)x_b^{-M_t}x_s^{S_t-M_t} + (1-\alpha)x_b^{B_t-M_t}x_s^{S_t-M_t}] \\ + B_t log(\epsilon_b+\mu) + S_t log(\epsilon_s+\mu) - (\epsilon_b+\epsilon_s) + M_t[log(x_b)+log(x_s)] - log(S_t!B_t!), \end{aligned} \tag{4}$$

where $M_t = min(B_t, S_t) + max(B_t, S_t)/2$, $x_b = \epsilon_b/(\mu+\epsilon_b)$ and $x_s = \epsilon_s/(\mu+\epsilon_s)$.

Lin and Ke (2011) introduce another algebraically equivalent factorization of the equation (2), $L_{LK}(\Theta|T) \equiv \sum_{t=1}^{T} L_{LK}(\Theta|B_t, S_t)$ where

$$\begin{aligned} L_{LK}(\Theta|B_t, S_t) = log[\alpha\delta exp(e_{1t}-e_{maxt}) + \alpha(1-\delta)exp(e_{2t}-e_{maxt}) + (1-\alpha)exp(e_{3t}-e_{maxt})] \\ + B_t log(\epsilon_b+\mu) + S_t log(\epsilon_s+\mu) - (\epsilon_b+\epsilon_s) + e_{maxt} - log(S_t!B_t!), \end{aligned} \tag{5}$$

where $e_{1t} = -\mu - B_t log(1+\mu/\epsilon_b)$, $e_{2t} = -\mu - S_t log(1+\mu/\epsilon_s)$, $e_{3t} = -B_t log(1+\mu/\epsilon_b) - S_t log(1+\mu/\epsilon_s)$ and $e_{maxt} = max(e_{1t}, e_{2t}, e_{3t})$. The last term $log(S_t!B_t!)$ is constant with respect to the parameter vector Θ, and is, therefore, dropped in the MLE for both factorizations.

Boundary Solutions

Another source of bias in estimating PIN arises from boundary solutions. Yan and Zhang (2012) indicate that in calculating PIN, parameter estimates $\hat{\alpha}$ and $\hat{\delta}$ usually fall onto the boundaries of the parameter space, that is, they are equal to zero or one. PIN estimate presented in equation (3) is directly related to the estimate of $\hat{\alpha}$. Letting $\hat{\alpha}$ equal to zero will make sure that PIN is zero as well. This can create a sample selection bias in portfolio formation, especially for quarterly estimations[4]. Yan and Zhang (2012) show that;

$$E(B) = \alpha(1-\delta)\mu + \epsilon_b \tag{6}$$

$$E(S) = \alpha\delta\mu + \epsilon_s \tag{7}$$

Then, they propose the following algorithm to overcome the bias created due to boundary solutions. Let $(\alpha^0, \delta^0, \epsilon_b^0, \epsilon_s^0, \mu^0)$ be the initial parameter function to be placed in the non-linear program presented in equation (4). In addition, let $\bar{B}$ and $\bar{S}$ be the average number of buy and sell orders.

$$\alpha^0 = \alpha_i, \quad \delta^0 = \delta_j, \quad \epsilon_b^0 = \gamma_k\bar{B}, \quad \mu^0 = \frac{\bar{B}-\epsilon_b^0}{\alpha^0(1-\delta^0)} \text{ and } \epsilon_s^0 = \bar{S} - \alpha^0\delta^0\mu^0 \tag{8}$$

where $\alpha_i, \delta_j, \gamma_k \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. This will yield 125 different PIN estimates along with their likelihood values. In line with Yan and Zhang (2012), we drop any initial parameter vector having negative values for ϵ_s^0. In addition, following Ersan and Alici (2016), we also drop any initial parameter vector with $\mu^0 > max(B_t, S_t)$. Yan and Zhang (2012) then select the estimate with non-boundary parameters yielding highest likelihood value. This method, by construction, spans the parameter space and tries to avoid local optima and provides non-boundary estimates for α.

[3]For example, Zagaglia (2012) provides a sample data to calculate PIN. In sample data the maximum trade number is 19. If you multiply each observation in the sample data by 10, the `pin_likelihood()` function of **FinAsym** package fails to provide results with the sample initial parameter vector.

[4]For quarterly estimations of PIN, one can be sure that there is at least one information event, earnings announcement. Therefore $\hat{\alpha}$ cannot be equal to zero.

Clustering Approach

In recent years, clustering algorithms are increasingly becoming popular in estimating the probability of informed trading due to efficiency concerns. Gan et al. (2015) and Ersan and Alici (2016) use clustering algorithms to estimate PIN. Gan et al. (2015) introduce a method that clusters the data into three groups (good news, bad news, no news) based on the mean absolute difference in order imbalance. Let $X_t = B_t - S_t$ be the order imbalance on day t computed as the difference between buy orders and sell orders. The clustering is then based on the distance function defined as $D(I, J) = |X_i - X_j|, \quad 1 \leq i, j \leq T$ where $i \neq j$. They use hierarchical agglomerative clustering (HAC) to group the data elements based on the distance matrix. Specifically, they use `hclust()` function of Müllner (2013) in R[5]. The algorithm sequentially clusters, in a bottom-up fashion, each observation into groups based on X_t and stops when it reaches three clusters. The theoretical framework of Easley et al. (1996) indicates that a stock has high (low) X_t on good (bad) days. Therefore, the cluster which has the highest (lowest) mean X_t is labelled as good (bad) news. The remaining cluster is then labelled as no news. Once each observation is grouped into their respective clusters (good news, bad news, no news), $c \in \{G, B, N\}$, the parameter estimates for $\Theta \equiv \{\alpha, \delta, \mu, \epsilon_b, \epsilon_s\}$ are calculated simply by counting. Let ω_c be the proportion of cluster c occupying the total number of days T, such that $\sum_{c=1}^{3} \omega_c = 1$. Similarly, let $\bar{B_c}$ and $\bar{S_c}$ be the average number of buys and sells on cluster c, respectively.

Then, the probability of an information event is given by $\hat{\alpha} = \omega_B + \omega_G$. Moreover, the estimate for the probability of information event releasing bad news is given by $\hat{\delta} = \omega_B / \hat{\alpha}$. The estimate for the arrival rate of buy orders of uninformed traders represented by $\hat{\epsilon_b} = \frac{\omega_B}{\omega_B + \omega_N} \bar{B_B} + \frac{\omega_N}{\omega_B + \omega_N} \bar{B_N}$. Similarly, the estimate for the arrival rate of sell orders of uninformed traders represented by $\hat{\epsilon_s} = \frac{\omega_G}{\omega_G + \omega_N} \bar{S_G} + \frac{\omega_N}{\omega_G + \omega_N} \bar{B_N}$. Finally, the arrival rate for the informed investors is calculated as $\hat{\mu} = \frac{\omega_G}{\omega_B + \omega_G} (\bar{B_G} - \hat{\epsilon_b}) + \frac{\omega_B}{\omega_B + \omega_G} (\bar{S_B} - \hat{\epsilon_s})$ where $(\bar{B_G} - \hat{\epsilon_b})$ corresponds to the buy rate of informed investors $\hat{\mu_b}$ and $(\bar{S_B} - \hat{\epsilon_s})$ corresponds to the sell rate of informed investors $\hat{\mu_s}$[6].

Through simulations, Gan et al. (2015) show that estimates calculated as above are proper candidates for the initial parameter values to be used in MLE process. Ersan and Alici (2016) argue that the estimates for the informed arrival rate, μ, contains a downward bias with GAN algorithm[7]. This is what we observe in this study as well. In addition, they state that GAN algorithm provides inaccurate estimates for δ. In order to overcome these issues, instead of using X_t, Ersan and Alici use absolute daily order imbalance, $|X_t|$, to cluster the data. They initially cluster, $|X_t|$ into two, again by using hclust(). The cluster with the lower mean daily absolute order imbalance is labelled as "no event" cluster and the remaining as "event" cluster. Then, the formation of "good" and "bad" event day clusters are obtained through separating the days in the "event" cluster into two with respect to the *sign* of the daily order imbalances. The parameter estimates are then computed with the same procedure presented above[8].

The InfoTrad Package

The R package **InfoTrad** provides five different functions `EHO()`,`LK()`,`YZ()`,`GAN()` and `EA()`. The first two functions provide likelihood specifications whereas the last three functions can be used to obtain parameter estimates for Θ to calculate PIN in equation (3). All five functions require a data frame that contains B_t in the first column, and S_t in the second column. We create B_t and S_t for ten hypothetical trading days[9]. `EHO()` and `LK()` read (B_t, S_t) and return the related functional form of the negative log likelihood. These objects can be used in any optimization procedure such as `optim()` to obtain the parameter estimates $\hat{\Theta} \equiv \{\hat{\alpha}, \hat{\delta}, \hat{\mu}, \hat{\epsilon_b}, \hat{\epsilon_s}\}$, the likelihood value and other specifications, in one iteration with a pre-specified initial value vector, Θ_0, for parameters. We define `EHO()` and `LK()` as simple likelihood specifications rather than functions that execute the MLE procedure. This is due to the fact that MLE estimators vary depending on the optimization procedure. Users who wish to develop alternative estimation techniques, based on the proposed likelihood factorization, can use `EHO()` and `LK()`. This is the underlying reason why those functions do not have built-in optimization procedures.

[5] `hclust()` function is used at its default setting in line with Gan et al. (2015).

[6] Both Gan et al. (2015) and Ersan and Alici (2016) do not mention the case where $\hat{\mu_b} < 0$ or $\hat{\mu_s} < 0$. It is fair to assume that in such cases, informed investors are not present on the buy (sell) side. Therefore, we set μ_b and μ_s equal to zero when we obtain a negative estimate.

[7] We also show that estimates for μ contains a significant downward bias due to poor choice of initial parameter value μ_0 when GAN algorithm is used.

[8] Ersan and Alici (2016) also provide an iterative process in which they systematically update the clusters. We plan to introduce this methodology in the future versions of our package.

[9] The numbers are randomly selected. We set numbers to be high enough so that the original likelihood framework presented in equation (1) cannot be used due to FPE. Easley et al. (1996) indicate that at least 60 days worth of data is required in order to obtain proper convergence for $\widehat{PIN}$. We use ten days for demonstration purposes.

By specifying `EHO()` and `LK()` as simple likelihood functions, we give developers the flexibility to select the most suitable optimization procedure for their application.

For researchers who want to calculate an estimate of PIN, `YZ()`, `GAN()` and `EA()` functions have built-in optimization procedures. Those functions read a likelihood specification value along with data. Likelihood specification can be set either to "LK" or to "EHO" with "LK" being the default. All estimation functions use `neldermead()` function of **nloptr** package to conduct MLE with the specified factorization. GAN and EA functions also use `hclust()` function of Müllner (2013) to conduct clustering. The output of these three functions is an object that provides $\{\hat{\alpha}, \hat{\delta}, \hat{\mu}, \hat{\epsilon_b}, \hat{\epsilon_s}, f(\hat{\Theta}), \widehat{PIN}\}$, where $f(\hat{\Theta})$ represents the optimal likelihood value given the parameter estimates $\hat{\Theta}$.

EHO() function

An example is provided below for `EHO()` with a sample data and initial parameter values. Notice that the first column of sample data is for B_t and second column is for S_t. Similarly, the initial parameter values are constructed as; $\Theta_0 = \{\alpha, \delta, \mu, \epsilon_b, \epsilon_s\}$. We use `optim()` with 'Nelder-Mead' method to execute MLE, however *developer* is flexible to use other methods as well.

```
library(InfoTrad)
# Sample Data
#   Buy Sell
#1  350  382
#2  250  500
#3  500  463
#4  552  550
#5  163  200
#6  345  323
#7  847  456
#8  923  342
#9  123  578
#10 349  455

Buy<-c(350,250,500,552,163,345,847,923,123,349)
Sell<-c(382,500,463,550,200,323,456,342,578,455)
data=cbind(Buy,Sell)

# Initial parameter values
# par0 = (alpha, delta, mu, epsilon_b, epsilon_s)
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("Nelder-Mead"), hessian = FALSE)

## Parameter Estimates
model$par[1] # Estimate for alpha
# [1] 0.9111102
model$par[2] # Estimate for delta
#[1] 0.0001231429
model$par[3] # Estimate for mu
# [1] 417.1497
model$par[4] # Estimate for eb
# [1] 336.075
model$par[5] # Estimate for es
# [1] 466.2539

## Estimate for PIN
(model$par[1]*model$par[3])/((model$par[1]*model$par[3])+model$par[4]+model$par[5])
# [1] 0.3214394
####
```

In this example, B_t and S_t vectors are selected so that the likelihood function cannot be represented as in equation (1). We set the initial parameters to be Θ_0=(0.5,0.5,300,400,500). For the given B_t, S_t and Θ_0 vectors, PIN measure is calculated as 0.32 with EHO factorization.

LK() function

An example is provided below for `LK()` function with a sample data and initial parameter values. Notice that the first column of sample data is for B_t and second column is for S_t. Similarly, the initial parameter values are constructed as; $\Theta_0 = \{\alpha, \delta, \mu, \epsilon_b, \epsilon_s\}$. We use `optim()` with 'Nelder-Mead' method to execute MLE, however *developer* is flexible to use other methods as well.

```
library(InfoTrad)
# Sample Data
#   Buy Sell
#1  350  382
#2  250  500
#3  500  463
#4  552  550
#5  163  200
#6  345  323
#7  847  456
#8  923  342
#9  123  578
#10 349  455

Buy<-c(350,250,500,552,163,345,847,923,123,349)
Sell<-c(382,500,463,550,200,323,456,342,578,455)
data=cbind(Buy,Sell)

# Initial parameter values
# par0 = (alpha, delta, mu, epsilon_b, epsilon_s)
par0 = c(0.5,0.5,300,400,500)

# Call LK function
LK_out = LK(data)
model = optim(par0, LK_out, gr = NULL, method = c("Nelder-Mead"), hessian = FALSE)

## The structure of the model output ##
model

#$par
#[1]   0.480277   0.830850 315.259805 296.862318 400.490830

#$value
#[1] -44343.21

#$counts
#function gradient
#    502       NA

#$convergence
#[1] 1

#$message
#NULL

## Parameter Estimates
model$par[1] # Estimate for alpha
# [1] 0.480277
model$par[2] # Estimate for delta
# [1] 0.830850
model$par[3] # Estimate for mu
# [1] 315.259805
model$par[4] # Estimate for eb
# [1] 296.862318
model$par[5] # Estimate for es
# [1] 400.4908

## Estimate for PIN
```

```
(model$par[1]*model$par[3])/((model$par[1]*model$par[3])+model$par[4]+model$par[5])
# [1] 0.178391
####
```

For the given B_t, S_t and Θ_0 vectors, PIN measure is calculated as 0.18 with LK factorization.

YZ() function

An example is provided below for `YZ()` function with a sample data. Notice that the first column of sample data is for B_t and second column is for S_t. In addition, the first example is with default likelihood specification LK and the second one is with EHO. Notice that `YZ()` function do not require any initial parameter vector Θ_0.

```
library(InfoTrad)
# Sample Data
#   Buy Sell
#1  350  382
#2  250  500
#3  500  463
#4  552  550
#5  163  200
#6  345  323
#7  847  456
#8  923  342
#9  123  578
#10 349  455

Buy<-c(350,250,500,552,163,345,847,923,123,349)
Sell<-c(382,500,463,550,200,323,456,342,578,455)
data<-cbind(Buy,Sell)

# Parameter estimates using the LK factorization of Lin and Ke (2011)
# with the algorithm of Yan and Zhang (2012).
# Default factorization is set to be "LK"

result=YZ(data)
print(result)

# Alpha: 0.3999999
# Delta: 0
# Mu: 442.1667
# Epsilon_b: 263.3333
# Epsilon_s: 424.9
# Likelihood Value: 44371.84
# PIN: 0.2004457

# Parameter estimates using the EHO factorization of Easley et. al. (2010)
# with the algorithm of Yan and Zhang (2012).

result=YZ(data,likelihood="EHO")
print(result)

# Alpha: 0.9000001
# Delta: 0.9000001
# Mu: 489.1111
# Epsilon_b: 396.1803
# Epsilon_s: 28.72002
# Likelihood Value: Inf
# PIN: 0.3321033
```

For the given B_t and S_t vectors, PIN measure is calculated as 0.20 with YZ algorithm along with LK factorization. Moreover, PIN measure is calculated as 0.33 with YZ algorithm along with EHO factorization.

GAN() function

An example is provided below for `GAN()` function with a sample data. Notice that the first column of sample data is for B_t and second column is for S_t. In addition, the first example is with default likelihood specification LK and the second one is with EHO. Notice that `GAN()` function do not require any initial parameter vector Θ_0.

```
library(InfoTrad)
# Sample Data
#   Buy Sell
#1  350  382
#2  250  500
#3  500  463
#4  552  550
#5  163  200
#6  345  323
#7  847  456
#8  923  342
#9  123  578
#10 349  455

Buy<-c(350,250,500,552,163,345,847,923,123,349)
Sell<-c(382,500,463,550,200,323,456,342,578,455)
data<-cbind(Buy,Sell)

# Parameter estimates using the LK factorization of Lin and Ke (2011)
# with the algorithm of Gan et. al. (2015).
# Default factorization is set to be "LK"

result=GAN(data)
print(result)

# Alpha: 0.3999998
# Delta: 0
# Mu: 442.1667
# Epsilon_b: 263.3333
# Epsilon_s: 424.9
# Likelihood Value: 44371.84
# PIN: 0.2044464

# Parameter estimates using the EHO factorization of Easley et. al. (2010)
# with the algorithm of Gan et. al. (2015)

result=GAN(data, likelihood="EHO")
print(result)

# Alpha: 0.3230001
# Delta: 0.4780001
# Mu: 481.3526
# Epsilon_b: 356.6359
# Epsilon_s: 313.136
# Likelihood Value: Inf
# PIN: 0.1884001
```

For the given B_t and S_t vectors, PIN measure is calculated as 0.20 with GAN algorithm along with LK factorization. Moreover, PIN measure is calculated as 0.19 with GAN algorithm along with EHO factorization.

EA() function

An example is provided below for `EA()` function with a sample data. Notice that the first column of sample data is for B_t and second column is for S_t. In addition, the first example is with default likelihood specification LK and the second one is with EHO. Notice that `EA()` function do not require

any initial parameter vector Θ_0.

```
library(InfoTrad)
# Sample Data
#   Buy Sell
#1  350  382
#2  250  500
#3  500  463
#4  552  550
#5  163  200
#6  345  323
#7  847  456
#8  923  342
#9  123  578
#10 349  455

Buy=c(350,250,500,552,163,345,847,923,123,349)
Sell=c(382,500,463,550,200,323,456,342,578,455)
data=cbind(Buy,Sell)

# Parameter estimates using the LK factorization of Lin and Ke (2011)
# with the modified clustering algorithm of Ersan and Alici (2016).
# Default factorization is set to be "LK"

result=EA(data)
print(result)

# Alpha: 0.9511418
# Delta: 0.2694005
# Mu: 76.7224
# Epsilon_b: 493.7045
# Epsilon_s: 377.4877
# Likelihood Value: 43973.71
# PIN: 0.07728924


# Parameter estimates using the EHO factorization of Easley et. al. (2010)
# with the modified clustering algorithm of Ersan and Alici (2016).

result=EA(data,likelihood="EHO")
print(result)

# Alpha: 0.9511418
# Delta: 0.2694005
# Mu: 76.7224
# Epsilon_b: 493.7045
# Epsilon_s: 377.4877
# Likelihood Value: 43973.71
# PIN: 0.07728924
```

For the given B_t and S_t vectors, PIN measure is calculated as 0.08 with EA algorithm along with LK factorization. Moreover, PIN measure is calculated, again, as 0.08 with EA algorithm along with EHO factorization.

Simulations and Performance Evaluation

In this section, we investigate the performance of the estimates obtained for Θ and PIN using the existing methods. We evaluate the methods based on their accuracy proxied by mean absolute errors (MAE)[10]. We first examine how the estimates vary in different trade intensity levels. To this end, we follow the methodology in Gan et al. (2015). Let I be the the set of trade intensity levels ranging from 50 to 5000 at step size of 50, that is, I=$\{50, 100, 150, \ldots, 5000\}$. We first set our parameters as

[10] All estimations are conducted on a 2.6 Intel i7-6700HQ CPU.We do not consider speed as a performance measure since the average processing time for each method is less than 10 seconds.

$\Theta = \{\alpha = 0.5, \delta = 0.5, \mu = 0.2i, \epsilon_b = 0.4i, \epsilon_s = 0.4i\}$, where $i \in I$. For each trade intensity level, we generate N=50 random samples of $\tilde{\alpha}$ and $\tilde{\delta}$ that are binomially distributed with parameters α and δ respectively. $\tilde{\alpha}$ and $\tilde{\delta}$ proxy the content of the information event. For each pair of $\tilde{\alpha}$, $\tilde{\delta}$ values, we generate buy and sell values (B_t, S_t) for hypothetical T=60 days in the following manner;

- if $\tilde{\alpha}$ = 0, then there is no information event, therefore, generate $B_t \sim Pois(\epsilon_b)$ and $S_t \sim Pois(\epsilon_s)$.
- if $\tilde{\alpha}$ = 1, and $\tilde{\delta}$ =1, then there is bad news, therefore generate $B_t \sim Pois(\epsilon_b)$ and $S_t \sim Pois(\epsilon_s + \mu)$
- if $\tilde{\alpha}$ = 1, and $\tilde{\delta}$ =0, then there is good news, therefore generate $B_t \sim Pois(\epsilon_b + \mu)$ and $S_t \sim Pois(\epsilon_s)$

We then form the joint likelihood function represented by equation (4) in EHO form or by equation (5) in LK form and obtain the estimates using `YZ()`, `GAN()` or `EA()` methods.

The results are presented in Table 1 which indicates that `YZ()` method with `LK()` factorization provides the PIN estimates with lowest MAE. Although the clustering algorithms, especially `GAN()` method, provide powerful estimates of $\hat{\alpha}, \hat{\delta}, \hat{\epsilon_b}, \hat{\epsilon_s}$, they fail to estimate the arrival rate of informed investors $\hat{\mu}$,accurately. This is in line with Ersan and Alici (2016). On the contrary, `YZ()` method with `EHO()` factorization provides the best estimates for $\hat{\mu}$, but fails to provide good estimates for other parameters.

Method	Factorization	$\widehat{PIN}$	$\hat{\alpha}$	$\hat{\delta}$	$\hat{\mu}$	$\hat{\epsilon_b}$	$\hat{\epsilon_s}$
YZ	**LK**	0.075	0.199	0.059	415.2	104.3	109.0
YZ	**EHO**	0.134	0.428	0.310	154.6	288.3	247.4
GAN	**EHO**	0.101	0.087	0.083	479.4	124.1	117.3
GAN	**LK**	0.101	0.087	0.083	479.5	123.8	118.1
EA	**LK**	0.102	0.268	0.274	484.6	128.7	119.3
EA	**EHO**	0.102	0.270	0.275	483.1	128.5	107.8

Table 1: This table represents the mean absolute errors (MAE) of the parameter estimates obtained by a given method for a given factorization. Each row represents a different method with a different factorization. First two column represent the specification of method and factorization respectively. The last six columns represents the power of estimates of PIN along with the parameter space $\Theta \equiv \{\alpha, \delta, \mu, \epsilon_b, \epsilon_s\}$. MAE measures for the estimates calculated as $\sum_{i=1}^{N} \frac{|\widehat{\Theta}_i - \Theta_i^{TR}|}{N}$ where $\widehat{\Theta}$ represent the estimates and Θ^{TR} represents the true value.

A more general way of examining the accuracy of PIN estimates is proposed in several studies (e.g, Lin and Ke (2011), Gan et al. (2015), Ersan and Alici (2016)). In this setting, we fix the trade intensity, I=2500. The total trade intensity represents the overall presence of informed and uninformed traders, that is, I=(μ, ϵ_b, ϵ_s). We then generate three probability terms p_1, p_2, p_3 with N=5000 random observations that are distributed uniformly between 0 and 1. p_1 represents the fraction of informed investors in total trade intensity, that is, μ=$p_1 * I$. The rest of the trade intensity is distributed equally to buy and sell orders of uninformed investors, that is, $e_b = e_s = (1 - p_1) * I/2$. p_2 represents the true parameter for the probability of news arrival, α, and p_3 is the true parameter for the content of the news, δ. We generate observations for $\tilde{\alpha}$ and $\tilde{\delta}$, as described earlier. For each pair of $\tilde{\alpha}$ and $\tilde{\delta}$, we generate buy and sell values (B_t, S_t) for hypothetical T=60 days, again, in the manner presented above; form the likelihood and obtain the parameter estimates.

The results are presented in Table 2. Similar to first simulation, `GAN()` captures the true nature of $\hat{\alpha}$ and $\hat{\delta}$ better than any other method with both factorizations. `YZ()` method with `EHO()` factorization performs best when estimating the arrival of informed traders, $\hat{\mu}$. The importance of estimating $\hat{\mu}$ becomes quite evident in Table 2. Although other methods outperform `YZ()` method with `EHO()` factorization in estimating α, ϵ_b and ϵ_s, it provides the best estimate for PIN due to it's performance on estimating $\hat{\mu}$.

Summary

This paper provides a short survey on five most widely used estimation techniques for the probability of informed trading (PIN) measure. In this paper, we introduce the R package **InfoTrad**, covering estimation procedures for PIN using EHO, LK factorizations along with YZ, GAN and EA algorithms (`EHO()`,`LK()`, `YZ()`, `GAN()` `EA()`). The functions `EHO()` and `LK()` read a (Tx2) matrix where the rows of the first column contains total number of buy orders on a given trading day t, B_t, and the rows

Method	Factorization	$\widehat{PIN}$	$\hat{\alpha}$	$\hat{\delta}$	$\hat{\mu}$	$\hat{\epsilon_b}$	$\hat{\epsilon_s}$
YZ	**LK**	0.323	0.428	0.432	1,212.0	303.4	325.0
YZ	**EHO**	0.237	0.437	0.357	942.9	386.0	470.2
GAN	**LK**	0.348	0.380	0.410	1,218.7	314.5	323.3
GAN	**EHO**	0.347	0.357	0.397	1,216.2	328.5	339.5
EA	**LK**	0.348	0.437	0.421	1,224.0	325.1	336.3
EA	**EHO**	0.347	0.428	0.413	1,222.0	331.3	345.9

Table 2: This table represents the mean absolute errors (MAE) of the parameter estimates obtained by a given method for a given factorization. Each row represents a different method with a different factorization. First two column represent the specification of method and factorization respectively. The last six columns represents the power of estimates of PIN along with the parameter space $\Theta \equiv \{\alpha, \delta, \mu, \epsilon_b, \epsilon_s\}$. MAE measures for the estimates calculated as $\sum_{i=1}^{N} \frac{|\widehat{\Theta}_i - \Theta_i^{TR}|}{N}$ where $\widehat{\Theta}$ represent the estimates and Θ^{TR} represents the true value.

of the second column contains the total number of sell orders on a given trading day t, S_t, where $t \in \{1, 2, \ldots, T\}$. In addition, they also require an initial parameter vector in the form of, $\Theta_0 = \{\alpha, \delta, \mu, \epsilon_b, \epsilon_s\}$. Both functions produce the respective log-likelihood functions.

The functions `YZ()`, `GAN()` and `EA()` read (B_t, S_t) as an input along with a likelihood specification that is set to '`LK`' by default. These functions do not require initial parameter matrix to obtain the parameter estimates when calculating PIN. All three functions use `neldermead()` method of **nloptr** as built-in optimization procedure for MLE. `YZ()` `GAN()` and `EA()` produce an object that gives the parameter estimates $\hat{\Theta}$ along with likelihood value and $\widehat{PIN}$.

Acknowledgments

This research is supported by the Scientific and Technological Research Council of Turkey (TUBITAK), Grant Number: 116K335.

Bibliography

N. Aktas, E. De Bodt, F. Declerck, and H. Van Oppens. The PIN anomaly around *m&a* announcements. *Journal of Financial Markets*, 10(2):169–191, 2007. URL https://doi.org/10.1016/j.finmar.2006.09.003. [p31]

J. Duarte and L. Young. Why is PIN priced? *Journal of Financial Economics*, 91(2):119–138, 2009. URL https://doi.org/10.1016/j.jfineco.2007.10.008. [p31]

D. Easley, N. M. Kiefer, M. O'Hara, and J. B. Paperman. Liquidity, information, and infrequently traded stocks. *The Journal of Finance*, 51(4):1405–1436, 1996. URL https://doi.org/10.1111/j.1540-6261.1996.tb04074.x. [p31, 32, 34]

D. Easley, M. O'Hara, and J. Paperman. Financial analysts and information-based trade. *Journal of Financial Markets*, 1(2):175–201, 1998. URL https://doi.org/10.1016/s1386-4181(98)00002-0. [p31]

D. Easley, M. O'Hara, and G. Saar. How stock splits affect trading: A microstructure approach. *Journal of Financial and Quantitative Analysis*, 36(01):25–51, 2001. URL https://doi.org/10.2307/2676196. [p31]

D. Easley, S. Hvidkjaer, and M. O'Hara. Is information risk a determinant of asset returns? *The Journal of Finance*, 57(5):2185–2221, 2002. URL https://doi.org/10.1111/1540-6261.00493. [p31, 32]

D. Easley, S. Hvidkjaer, and M. O'Hara. Factoring information into returns. *Journal of Financial and Quantitative Analysis*, 2010. URL https://doi.org/10.1017/s0022109010000074. [p31, 33]

A. Ellul and M. Pagano. IPO underpricing and after-market liquidity. *Review of Financial Studies*, 19(2):381–421, 2006. URL https://doi.org/10.1093/rfs/hhj018. [p31]

O. Ersan and A. Alici. An unbiased computation methodology for estimating the probability of informed trading (PIN). *Journal of International Financial Markets, Institutions and Money*, 43:74–94, 2016. URL https://doi.org/10.1016/j.intfin.2016.04.001. [p31, 32, 33, 34, 40]

T. Foucault, M. Pagano, and A. Röell. *Market Liquidity: Theory, Evidence, and Policy*. Oxford University Press, 2013. URL https://doi.org/10.1093/acprof:oso/9780199936243.001.0001. [p32]

Q. Gan, W. C. Wei, and D. Johnstone. A faster estimation method for the probability of informed trading using hierarchical agglomerative clustering. *Quantitative Finance*, 15(11):1805–1821, 2015. URL https://doi.org/10.1080/14697688.2015.1023336. [p31, 32, 34, 39, 40]

C. Lee and M. J. Ready. Inferring trade direction from intraday data. *The Journal of Finance*, 46(2): 733–746, 1991. URL https://doi.org/10.1111/j.1540-6261.1991.tb02683.x. [p31]

H.-W. W. Lin and W.-C. Ke. A computing bias in estimating the probability of informed trading. *Journal of Financial Markets*, 14(4):625–640, 2011. URL https://doi.org/10.1016/j.finmar.2011.03.001. [p31, 33, 40]

A. J. Menkveld. Splitting orders in overlapping markets: A study of cross-listed stocks. *Journal of Financial Intermediation*, 17(2):145–174, 2008. URL https://doi.org/10.1016/j.jfi.2007.05.004. [p32]

D. Müllner. Fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software*, 53(9):1–18, 2013. URL https://doi.org/10.18637/jss.v053.i09. [p34, 35]

E. R. Odders-White and M. J. Ready. Credit ratings and stock liquidity. *Review of Financial Studies*, 19 (1):119–157, 2006. URL https://doi.org/10.1093/rfs/hhj004. [p31]

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2016. [p32]

Y. Yan and S. Zhang. An improved estimation method and empirical properties of the probability of informed trading. *Journal of Banking & Finance*, 36(2):454–467, 2012. URL https://doi.org/10.1016/j.jbankfin.2011.08.003. [p31, 32, 33]

P. Zagaglia. *FinAsym*, 2012. URL https://CRAN.R-project.org/package=FinAsym. R package version 1.0. [p31, 32, 33]

P. Zagaglia. PIN: Measuring Asymmetric Information in Financial Markets with R. *The R Journal*, 5(1):80–86, 2013. URL https://journal.r-project.org/archive/2013/RJ-2013-008/index.html. [p31, 32]

Duygu Çelik
Bilkent University
Bilkent University Department of Management 06800 Bilkent Ankara
Turkey
duygu.celik@bilkent.edu.tr

Murat Tiniç
Bilkent University
Bilkent University Department of Management 06800 Bilkent Ankara
Turkey
tinic@bilkent.edu.tr