Applied Econometrics 3, Generalized hyperbolic distribution

Евгений Орлов

20.11.2014

### Данные

В качестве входных данных взяты значения закрытия индекса РТС в период с 01.01.2010 по 07.11.2014 включительно.

Источником данных выступила ИТС QUIK.

rts.df <- read.csv("RTSI [Price].txt", header=TRUE)  
tail(rts.df)

## X.TICKER. X.PER. X.DATE. X.TIME. X.OPEN. X.HIGH. X.LOW.  
## 2996 RTSI [RTSIDX] Daily 20141030 0 1059.62 1098.68 1040.95  
## 2997 RTSI [RTSIDX] Daily 20141031 0 1098.68 1112.71 1078.47  
## 2998 RTSI [RTSIDX] Daily 20141103 0 1091.44 1091.44 1073.30  
## 2999 RTSI [RTSIDX] Daily 20141105 0 1078.33 1078.33 1035.86  
## 3000 RTSI [RTSIDX] Daily 20141106 0 1054.45 1054.45 1007.64  
## 3001 RTSI [RTSIDX] Daily 20141107 0 1017.28 1028.20 978.62  
## X.CLOSE. X.VOL.  
## 2996 1098.68 23465  
## 2997 1091.44 23057  
## 2998 1078.33 16305  
## 2999 1054.45 20785  
## 3000 1017.28 20765  
## 3001 1010.73 20834

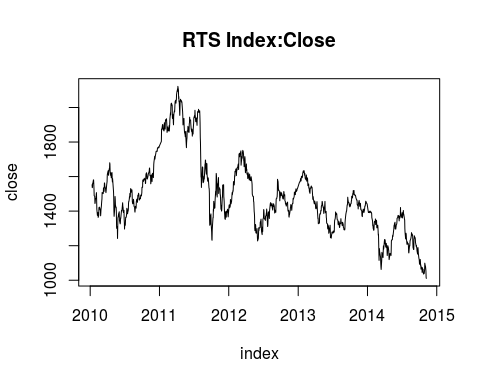
# Цены закрытия начиная с 01.01.2010 по 07.11.2014 включительно  
rts.close <- rts.df[rts.df$X.DATE. >= 20100101, "X.CLOSE."]  
# Длина выборки  
rcl <- length(rts.close)  
print(paste0("Размер выборки: ", rcl))

## [1] "Размер выборки: 1221"

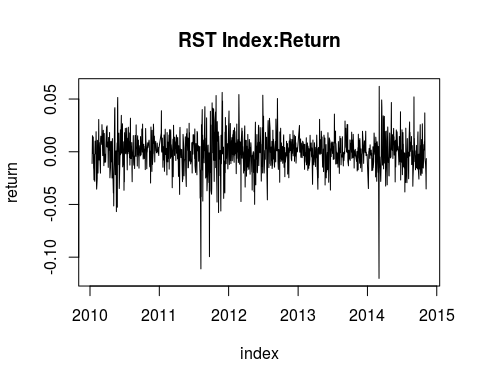
# Вектор доходностей  
rts.ret <- rts.close[2:rcl] / rts.close[1:(rcl-1)] - 1

Графики значений закрытия и доходностей:

# График значений закрытия  
plot(rts.dates, rts.close,   
 type='l', xlab='index', ylab='close', main='RTS Index:Close')



# График доходностей  
plot(rts.dates[-1], rts.ret,   
 type='l', xlab='index', ylab='return', main='RST Index:Return')



### Оценки риска на основе наилучшей модели по всей совокупности наблюдений

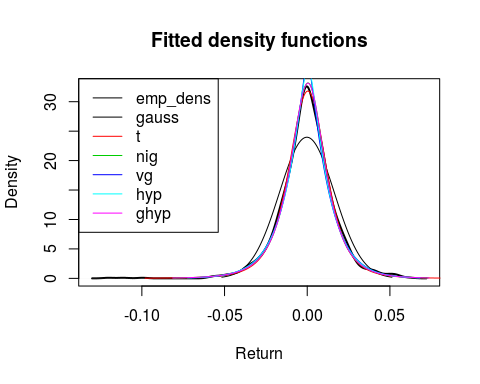
Для наглядного сравнения подберем оптимальные параметры для рассматриваемых распределений и отобразим их на графике.

library(ghyp, quietly=TRUE)

##   
## Attaching package: 'gplots'  
##   
## The following object is masked from 'package:stats':  
##   
## lowess

# Подгон параметров распределения  
# Эмпирическая плотность  
rts.emp\_dens<-density(rts.ret, bw='nrd')  
# Нормальное распределение  
rts.gauss <- fit.gaussuv(rts.ret)  
# t-распределение Стьюдента  
rts.t <- fit.tuv(rts.ret, silent=TRUE)  
# Нормально-обратное гауссовское  
rts.nig <- fit.NIGuv(rts.ret, silent=TRUE)  
# Variance-Gamma  
rts.vg <- fit.VGuv(rts.ret, silent=TRUE)  
# Гиперболическое  
rts.hyp <- fit.hypuv(rts.ret, silent=TRUE)  
# Обобщенное гиперболическое  
rts.ghyp <- fit.ghypuv(rts.ret, silent=TRUE)

# Визуализация подобранных распределений  
colors <- palette(rainbow(6))  
plot(rts.emp\_dens, type="l", lwd=2,   
 main="Fitted density functions", xlab='Return')  
lines(rts.gauss, type="l", col=colors[1])  
lines(rts.t, type="l", col=colors[2])  
lines(rts.nig, type="l", col=colors[3])  
lines(rts.vg, type="l", col=colors[4])  
lines(rts.hyp, type="l", col=colors[5])  
lines(rts.ghyp, type="l", col=colors[6])  
legend('topleft', c("emp\_dens", "gauss", "t", "nig", "vg", "hyp", "ghyp"),   
 col=c("black", colors), lty=rep(1, 7))



Выберем наилучшую модель распределения по всей выборке доходностей на основе критерия Акаике.

# Выбор наилучшей модели на основе критерия Акаике  
aic.uv <- stepAIC.ghyp(rts.ret, dist=c("gauss","t","hyp","ghyp","nig", "vg"),  
 silent=TRUE)

## Currently fitting: asymmetric t   
## Currently fitting: asymmetric hyp   
## Currently fitting: asymmetric ghyp   
## Currently fitting: symmetric t   
## Currently fitting: symmetric hyp   
## Currently fitting: symmetric ghyp   
## Currently fitting: gauss

rts.best <- aic.uv$best.model  
summary(rts.best) # Информация о наилучшей модели

## Symmetric Hyperbolic Distribution:  
##   
## Parameters:  
## alpha.bar mu sigma gamma   
## 3.093693e-01 4.422141e-05 1.639289e-02 0.000000e+00   
##   
## Call:  
## stepAIC.ghyp(data = rts.ret, dist = c("gauss", "t", "hyp", "ghyp", "nig", "vg"), silent = TRUE)  
##   
## Optimization information:  
## log-Likelihood: 3353.404   
## AIC: -6700.809   
## Fitted parameters: alpha.bar, mu, sigma; (Number: 3)  
## Number of iterations: 162   
## Converged: TRUE

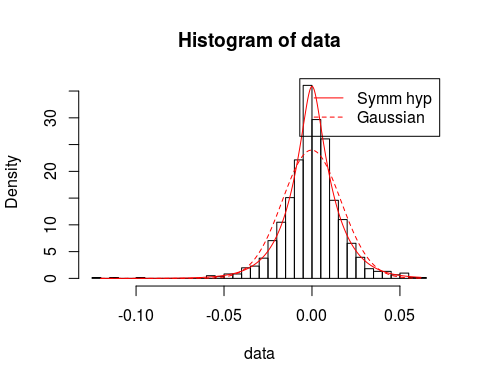
aic.uv$fit.table # Сводная информация о рассмотренных моделях

## model symmetric lambda alpha.bar mu sigma  
## 5 hyp TRUE 1.0000000 0.3093693 4.422141e-05 0.01639289  
## 6 ghyp TRUE -0.4757815 0.7790557 8.832965e-05 0.01653967  
## 4 t TRUE -1.8979326 0.0000000 7.979358e-05 0.01707020  
## 2 hyp FALSE 1.0000000 0.3424086 5.723108e-04 0.01634509  
## 3 ghyp FALSE -0.5807196 0.7736324 7.206229e-04 0.01652974  
## 1 t FALSE -1.9147803 0.0000000 7.125477e-04 0.01700290  
## 7 gauss TRUE NA Inf -2.124802e-04 0.01664162  
## gamma aic llh converged n.iter  
## 5 0.0000000000 -6700.809 3353.404 TRUE 162  
## 6 0.0000000000 -6699.836 3353.918 TRUE 275  
## 4 0.0000000000 -6699.720 3352.860 TRUE 116  
## 2 -0.0007866952 -6699.718 3353.859 TRUE 221  
## 3 -0.0009252824 -6699.122 3354.561 TRUE 476  
## 1 -0.0009412862 -6698.857 3353.429 TRUE 191  
## 7 0.0000000000 -6528.660 3266.330 TRUE 0

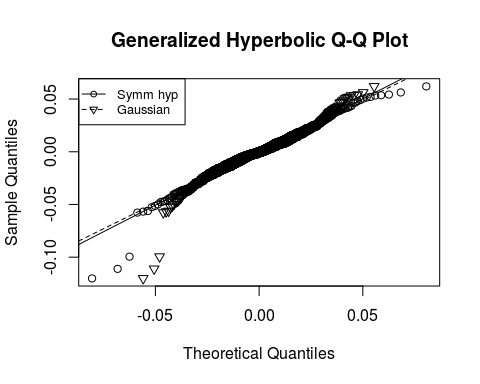
На основании критерия Акаике наилучшей моделью для распределения данных было выбрано симметричное гиперболическое распределение. Из таблицы видно, что в то время как для остальных распределений значения критерия Акаике и функции максимального правдоподобия примерно равны, нормальное распределение заметно уступает в качестве потенциальной модели распределения доходности индекса.

Визуально сравним подобранное симметричное гиперболическое распределение с нормальным.

# Гистограмма и график квантиль-квантиль   
hist(rts.best)



qqghyp(rts.best)



На верхней картинке наглядно видно, что подобранное симметричное гиперболическое распределение лучше соответствует выборке доходности.

Из графика "квантиль-квантиль" заметно, что

1. Хвост слева у распределения выборки толще, чем у нормального распределения. Гиперболическое распределение хорошо описывает хвост слева за исключением 3 значительных выбросов;
2. Хвост справа у распределения выборки толще, чем у нормального распределения, но тоньше, чем у гиперболического распределения.

Для верификации полученных результатов проведем тест Колмогорова-Смирнова для обоих распределений:

ks.test(rts.ret, rghyp(n=100\*length(rts.ret), object=rts.best))

## Warning in ks.test(rts.ret, rghyp(n = 100 \* length(rts.ret), object =  
## rts.best)): p-value will be approximate in the presence of ties

##   
## Two-sample Kolmogorov-Smirnov test  
##   
## data: rts.ret and rghyp(n = 100 \* length(rts.ret), object = rts.best)  
## D = 0.0171, p-value = 0.8696  
## alternative hypothesis: two-sided

ks.test(rts.ret, rghyp(n=100\*length(rts.ret), object=rts.gauss))

## Warning in ks.test(rts.ret, rghyp(n = 100 \* length(rts.ret), object =  
## rts.gauss)): p-value will be approximate in the presence of ties

##   
## Two-sample Kolmogorov-Smirnov test  
##   
## data: rts.ret and rghyp(n = 100 \* length(rts.ret), object = rts.gauss)  
## D = 0.0606, p-value = 0.0002834  
## alternative hypothesis: two-sided

p-value теста для гиперболического распределения (>> 0.50) указывает в пользу того, что гиперболическое распредение хорошо описывает распределение выборки доходностей.

В то же самое время p-value для нормального минимально (<< 0.01), что подтверждает идею о том, что нормальное распределение не подходит для описания распределения выборки.

Теперь проведем оценку рисков с помощью 1-дневных VaR и ES на основе всей совокупности наблюдений.

При расчете будем использовать следующие параметры:

# Параметры расчета VaR и ES  
conf.level <- 0.90 # доверительный уровень VaR  
alpha <- 1 - conf.level # уровень значимости  
N <- 10^6 # кол-во сгенерированных доходностей

Результаты получаем с помощью метода Монте-Карло:

# Симуляция и вывод результатов  
rts.best.sim <- sort(rghyp(n=N, object=rts.best))  
rts.gauss.sim <- sort(rghyp(n=N, object=rts.gauss))  
VaR1.best <- rts.best.sim[alpha\*N]  
VaR1.gauss <- rts.gauss.sim[alpha\*N]  
VaR2.best <- qghyp(alpha, object=rts.best)  
VaR2.gauss <- qghyp(alpha, object=rts.gauss)  
ES.best <- mean(rts.best.sim[1:(alpha\*N-1)])  
ES.gauss <- mean(rts.gauss.sim[1:(alpha\*N-1)])  
print(paste0("VaR1 best model/gauss: ", round(VaR1.best, 4),   
 "/", round(VaR1.gauss, 4)))

## [1] "VaR1 best model/gauss: -0.0187/-0.0216"

print(paste0("VaR2 best model/gauss: ", round(VaR2.best, 4),   
 "/", round(VaR2.gauss, 4)))

## [1] "VaR2 best model/gauss: -0.0188/-0.0215"

print(paste0("ES best model/gauss: ", round(ES.best, 4),   
 "/", round(ES.gauss, 4)))

## [1] "ES best model/gauss: -0.03/-0.0295"

На первый взгляд результат расчета VaR оказался неожиданным. Толщина хвостов у гиперболического распределения больше чем у нормального, но значение VaR оказалось меньшим. Я думаю, что дело здесь в выборе уровня значимости (alpha). Как наглядно заметно на графике, содержащем гистограмму, 10-й персентиль нормального распределения оказывается меньше соответствующего персентиля гиперболического распределения.

Учитывая этот результат, можно ожидать, что при построении кривой VaR для данного уровня значимости, VaR на основе нормального распределения будет систематически завышен.

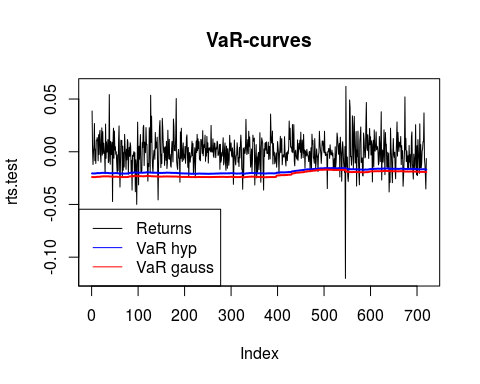
### Построение кривых VaR

При построении кривой VaR на основе гиперболического и нормального распределений параметры распределений переоцениваются каждый день, используя последние 500 известных значений доходности.

T1 <- 500  
T2 <- length(rts.ret)-T1  
VaR.hyp <- numeric() # VaR-кривая на основе гиперболического распределения  
VaR.gauss <- numeric() # VaR-кривая на основе нормального распределения  
  
# Построение VaR-кривых  
h <- T1 # длина обучающей выборки  
for (i in (T1+1):(T1+T2)) {  
 rts.train <- rts.ret[(i-h):(i-1)]  
 rts.hyp.fit <- fit.hypuv(rts.train, symmetric=TRUE, silent=TRUE)  
 rts.gauss.fit <- fit.gaussuv(rts.train)  
   
 VaR.hyp[i-T1] <- qghyp(alpha, object=rts.hyp.fit)  
 VaR.gauss[i-T1] <- qghyp(alpha, object=rts.gauss.fit)  
}

Полученные кривые VaR отображены на графике.

# График  
rts.test <- rts.ret[(T1+1):(T1+T2)]  
plot(rts.test, type="l", main="VaR-curves")  
lines(VaR.hyp, col="blue", lwd=2)  
lines(VaR.gauss, col="red", lwd=2)  
legend('bottomleft', c("Returns", "VaR hyp", "VaR gauss"),   
 col=c("black", "blue", "red"), lty=rep(1, 3))



Из графика видно, что предположения предыдущего пункта подтвердились, и при уровне значимости alpha равном 0.10 значение VaR на основе нормального распределения превышает значение VaR на основе симметричного гиперболического распределения.

### Сравнение результатов

Определим функции для теста Купика, функций потерь Лопеса и Бланко-Ила:

# Частота пробоев  
kupiec.test <- function(ret, VaR, alpha) {  
 # Тест Купика:  
 # H0: модельная и эмпирическая частоты пробоя VaR совпадают  
 K <- sum(ret < VaR)  
 T2 <- length(ret)  
 alpha0 <- K / T2  
 S <- -2\*log((1-alpha)^(T2-K) \* alpha^K) + 2\*log((1-alpha0)^(T2-K) \* alpha0^K)  
 p.value <- 1-pchisq(S, df=1)  
 return(c(alpha0, p.value))  
}  
  
# Глубина пробоев  
lopez.lf <- function(ret, VaR) {  
 # Функция потерь Лопеса  
 K <- sum(ret < VaR)  
 value <- sum((ret-VaR)^2\*(ret < VaR)) / K  
}  
  
blanco.lf <- function(ret, VaR) {  
 # Функция потерь Бланко-Ила  
 K <- sum(ret < VaR)  
 value <- sum((ret-VaR)/VaR \* (ret < VaR)) / K  
}

## [1] "Kupiec test, alpha = 0.1"

## [1] "hyp: alpha0 = 0.093056, p-value = 0.530187"

## [1] "gauss: alpha0 = 0.066667, p-value = 0.001605"

Из результатов теста Купика видно, что обе частоты превышений VaR не дотягивают до целевого значения в 10%. При этом результат у кривой VaR на основе гиперболического распределения оказался значительно лучше (p-value > 0.50 против p-value << 0.01).

## [1] "Lopez loss function hyp/gauss: 0.000279/0.00033"

## [1] "Blanco loss function hyp/gauss: 0.545237/0.52326"

В условиях, когда кривая VaR на основе нормального распределения не проходит тест Купика, анализ глубины превышений VaR теряет свою актуальность.

### Вывод

Для заданного значения уровня значимости alpha = 0.1, подбор распределения из семейства обобщенного гиперболического распределения даёт качественную модель для построения кривой VaR и тем самым приносит ощутимые результаты, так как нормальное распределение с задачей в данном случае не сравляется (VaR на основе нормального распределения систематически завышен, и VaR-кривая не проходит тест Купика).