finc584_ps6

William Lee

17 November, 2022

1 A

The naive point estimates are presented in part B for simplicity.

```
fred <- readxl::read_xlsx(file.path(proj, "FRED-QD.xlsx")) %>%
  select(pcecc96, sp500) %>%
  mutate(R_t = sp500/dplyr::lag(sp500,1),
         R_t1 = dplyr::lag(R_t,1),
         R_t2 = dplyr::lag(R_t,2),
         c_ratio = pcecc96/dplyr::lag(pcecc96,1),
         c_ratio1 = dplyr::lag(c_ratio,1),
         c_ratio2 = dplyr::lag(c_ratio,2)) %>%
  drop_na()

gmm.moments <- function(params, data = fred){
  beta <- params[1]
  gamma <- params[2]

  moment1 <- beta*(data['c_ratio']^(-gamma))*data['R_t'] - 1
  moment2 <- moment1 * data['R_t1']
  moment3 <- moment1 * data['R_t2']
  moment4 <- moment1 * data['c_ratio1']
  moment5 <- moment1 * data['c_ratio2']

  return(as.matrix(cbind(moment1, moment2, moment3, moment4, moment5)))
}

l2_metric <- function(x, weight_mat){
  return(colMeans(gmm.moments(x)) %*% weight_mat %*% colMeans(gmm.moments(x)))
}

naive <- optim(par = c(.7,2), fn = function(params)
  l2_metric(params, weight_mat = diag(5)), lower = c(0,0), upper = c(1, 20))
```

1 B

```
w_optimal <- solve(1/nrow(fred) * t(gmm.moments(naive$par)) %*% gmm.moments(naive$par))

optimal_params <- optim(par = naive$par, fn = function(params)
```

```
  l2_metric(params, weight_mat = w_optimal), lower = c(0,0), upper = c(1, 20))

kable(data.frame(var = c('beta', 'gamma'),
                 naive = naive$par, optimal = optimal_params$par), digits = 4)
```

var	naive	optimal
beta	0.9977	0.9964
gamma	1.9829	1.7953

1 C

```
gradient <- function(params,  data = fred){
  beta <- params[1]
  gamma <- params[2]

  # Derivative WRT beta
  e <- data[['R_t']]*data[['c_ratio']]^(-gamma)
  moment1_beta <- mean(e)
  moment2_beta <- mean(e * data[['R_t1']])
  moment3_beta <- mean(e * data[['R_t2']])
  moment4_beta <- mean(e * data[['c_ratio1']])
  moment5_beta <- mean(e * data[['c_ratio2']])

  # Derivative WRT gamma
  e <- -beta*data[['R_t']]*(data[['c_ratio']]^(-gamma))*log(data[['c_ratio']])
  moment_1_gamma <- mean(e)
  moment2_gamma <- mean(e * data[['R_t1']])
  moment3_gamma <- mean(e * data[['R_t2']])
  moment4_gamma <- mean(e * data[['c_ratio1']])
  moment5_gamma <- mean(e * data[['c_ratio2']])

  return(matrix(c(moment1_beta, moment2_beta, moment3_beta,
                  moment4_beta, moment5_beta, moment_1_gamma, moment2_gamma,
                  moment3_gamma, moment4_gamma, moment5_gamma), ncol = 5, byrow = T))

}

H <- gradient(optimal_params$par)

vcov_HC <- sqrt(diag((1/nrow(fred))* solve(H %*% w_optimal %*% t(H))))

ci.upper <- optimal_params$par + 1.96*(vcov_HC)
ci.lower <- optimal_params$par - 1.96*(vcov_HC)

kable(x = data.frame(var = c('beta', 'gamma'), optimal = optimal_params$par,
                     se = vcov_HC, ci.lower = ci.lower, ci.upper = ci.upper),
      digits = 3)
```

var	optimal	se	ci.lower	ci.upper
beta	0.996	0.018	0.961	1.032
gamma	1.795	2.021	-2.167	5.757

1 D

```
J <- nrow(fred) * colMeans(gmm.moments(optimal_params$par), w_optimal) %*%
  w_optimal %*% colMeans(gmm.moments(optimal_params$par), w_optimal)

J.Stat <- 1 - pchisq(J, df = 3)
```

The test statistic for over-identification is 25.8085555 which comes from a $\chi^2(3)$. The corresponding p-value is 1.0459445×10^{-5} which allows us to reject the null. The fact that instruments, which we think should have worked, are not satisfying the overidentification test make us question the validity of the model itself.