Sparse Covariance Estimation for Financial Time Series

Siddarth Viswanathan

Summary

Measurements each consisting of many variables arise in many scientific fields including biology, sociology, and finance. When using such multi-valued data it is vital to understand relations among these variables when constructing models, reducing the data, or studying underlying latent patterns. These variable relations are summarized in the covariance matrix, which is a fundamental tool in most statistical analyses. However when the number of variables in each measurement is much larger than the number of measurements the covariance matrix is difficult to estimate and requires novel methods. Using R and a financial dataset of 504 measurements of daily adjusted closing prices of 1547 medium-sized companies we explore, implement, and compare four methods for large covariance matrix estimation. The results do not indicate any clearly optimal method but provide insight into the immense difficulty of optimal high-dimensional covariance estimation.

Introduction

Covariance matrix estimation is fundamental in modern statistical analyses. However when the number of measured variables is much larger than the number of measurements the standard empirical covariance estimate has two main drawbacks [1]. First, the standard empirical covariance estimate is not invertible which implies an impossibility for calculating the precision matrix which is useful for understanding conditional dependencies among variables. Second, estimation errors accumulate rapidly leading to serious inaccuracies in the estimation.

A recent literature has explored many useful assumptions and computational approaches when estimating sparse, or high-dimensional, covariance matrices. Each method is based on various understanding of sparsity leading to many approaches including methods using thresholding which set small estimated values to zero, methods using penalized likelihoods, or methods using underlying factor approaches based on conditional sparsity [1, 2]. Methods are often based on Gaussian assumptions in the data though there are useful extensions to heavy-tailed distributions including elliptical distributions [1]. We explore applications of thresholding estimators to financial time series data.

Data Exploration and Preparation

We estimate sparse covariance matrices for data consisting of 504 daily adjusted closing prices for 1794 companies from the Russell 2000 index which is an index consisting of the 1000th-3000th largest U.S. companies [3]. The time series ranges from 01/02/2013 to 12/31/2014 which are years not known for abnormal market performance.

244 companies did not contain complete data for the time duration. These companies joined the Russell 2000 index at a later date. We remove these companies from the data since sparse covariance estimation is already difficult and erroneous with complete data and also the most-used missing-value sparse covariance or precision matrix estimation methods are based on a missing-at-random assumption which is not applicable to these financial time series [4]. Therefore our initial dataset consists of 504 daily adjusted closing prices for 1550 companies from the Russell 2000 index.

Figure 1 shows time series plots for all the stocks and reveals two series that are clearly outlying series, around 20 series which are somewhat outlying, and a vast majority of series with closing prices between 0 and 100. The series do not show trends of collectively increasing or decreasing in the time interval.

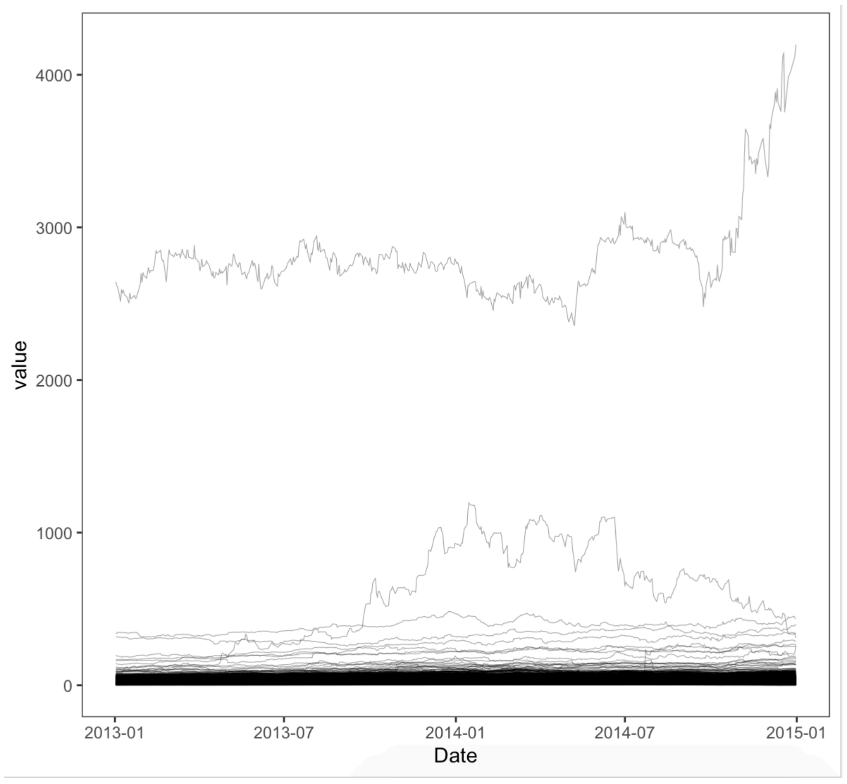

Figure 1 Adjusted daily closing price for 1550 companies from 01/02/2013-12/31/2014.

Figure 2 shows a correlation heatmap of 1550 stock prices. The heatmap shows the strong dependence we expect between the financial variables. Almost all correlations are close to positive one or close to negative one, there are very few intermediate shades. The extremity of this correlation heatmap along with the clearly outlying series in Figure 1 suggests a Sparse Principal Component Analysis to identify whether the data is strongly clustered and whether the few identified outliers are strongly impacting the identification of principal components.

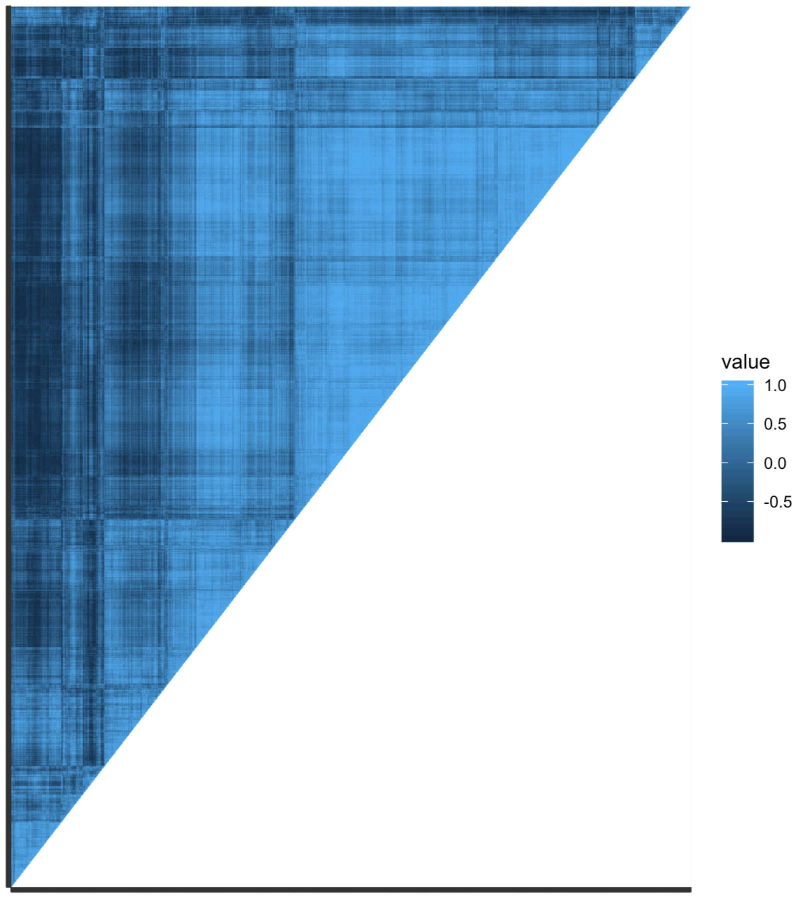

Figure 2 Correlation heatmap of 1550 stocks.

Sparse principal component analysis (SPCA) finds a weighting of the data using only few nonzero components [5]. The principal components are identified from a small subset of the covariates allowing for applicability to high-dimensional data. Robust sparse principal component methods reduce the impact of outliers while also dealing with high-dimensionality.

The purpose of using SPCA is to examine whether the data is formed from very few components as the heatmap tentatively suggests while also examining the effect of outliers on the formation of these components. Examining the number of outliers strongly impacting component formation will indicate whether these outlying series should be removed in the sparse covariance matrix estimation since it is known that both PCA and covariance estimation are sensitive to outliers.

Table 1 shows the cumulative proportion explained by the ranked principal components for various regimes of number of principal components over the full data and over data with series removed.

#PC	5-SPCA	10-RSPCA	10-RSPCA (3 series removed)	10-RSPCA (20 series removed)
1	0.49	0.5	0.63	0.61
2	0.91	0.906	0.79	0.78
3	0.947	0.947	0.85	0.84
4	0.963	0.962	0.88	0.88
5	0.97	0.97	0.91	0.89
6		0.975	0.92	0.91
7		0.978	0.93	0.92
8		0.981	94	0.93
9		0.982	0.945	0.937
10		0.983	0.95	0.94

Table 1 Cumulative proportion explained by various sparse principal component (SPCA) and robust sparse principal component (RSPCA) regimes.

Table 1 reveals four relevant findings. First, that the data are strongly explained by very few principal components. Second, that there is almost no difference between running RSPCA or SPCA. Third, that removing series has a strong impact on the number of relevant principal components. Fourth, that there is not much difference between removing three series and removing twenty series. Table 1 suggests that we should consider the three series with largest average daily closing price as outlying series and should remove these series before performing sparse covariance matrix estimation. The prepared data therefore consists of 504 daily adjusted closing prices for 1547 companies from the Russell 2000 index.

Sparse Covariance Matrix Estimation

If the data is multivariate normal we can use the method of [16] for quick and accurate estimation of a sparse covariance matrix. We perform three tests for multivariate normality of the data – Mardia, Henze-Zirkler, Royston—using the MVN R package [6]. Each of the three tests fails for size .05.

We proceed to covariance estimation using four methods with the CovTools and POET packages in R [7, 12] which do not require multivariate normal data – soft, hard, and adaptive thresholding methods, and the POET method.

The hard thresholding approach [8] is advantageous because it carries small computational burden, but is disadvantageous since it may lead to loss of positive definiteness. The method applies thresholding on off-diagonal elements of the sample covariance matrix which sets the sample matrix off-diagonal value to 0 if it is above a threshold value. The method is consistent with respect to the spectral norm.

The adaptive thresholding [9] has strong numerical and theoretical performance and is considered more optimal than the hard thresholding since the covariance matrix estimation can often have wide variability and therefore a thresholding approach with more flexibility will provide better estimation. The method is an adjustment of [8] to allow the threshold to be adaptive to each term of the covariance matrix.

Soft thresholding is also known to have no control of positive definiteness and therefore may be improved by hard-positive-definite and near-positive-definite covariance estimation approaches which guarantee a positive-definite covariance output natural to the covariance estimation problem [10, 11, 15]. Soft-thresholding is advantageous over hard-thresholding in that like adaptive thresholding it allows for threshold variation. Soft thresholding takes off-diagonal elements of the sample covariance matrix and adjusts them using the sample covariance matrix index sign and a variable threshold. In general hard, soft, and adaptive thresholding should be used in combination for optimal estimation.

The POET estimator generalizes the hard, soft, and adaptive approaches using principal orthogonal complement thresholding [12]. The method is optimization free but varies according to the number of factors with larger factors leading to robust performance.

Table 2 illustrates the sensitivity of the sparse covariance estimate to the choice of threshold. Using the absolute spectral norm distance we compare the covariance matrix estimate for adaptive- and hard-thresholding methods for various thresholds.

CovEst.Adaptive	**thr**	**0.01**	**0.3**	**0.6**	**0.9**
	0.01		3	501	9011
	0.3			499	9008
	0.6				8509
	0.9				

CovEst.hard	**thr**	**0.1**	**1**	**10**	**100**
	0.1		333	3884	26399
	1			3550	26066
	10				22515
	100				

Table 2 Absolute spectral norm distance between covariance matrix estimates for adaptive- and hard-thresholding methods for various thresholds. For example, the value of 501 in the cell at the meeting-point of .6 and .01 for adaptive-thresholding is the absolute spectral norm distance between the covariance matrix estimate for adaptive-thresholding with threshold as .6 and the covariance matrix estimate for adaptive-thresholding with threshold as .01.

Table 2 implies that threshold-estimators are extremely sensitive to the choice of threshold. For confidence in covariance matrix estimation quality we will need to experiment widely with threshold values since it is difficult to gather intuition for the optimal threshold. even when using the recommended heuristics guiding the choice of optimal threshold [8,9].

Comparison of Estimates and Discussion

We split the data into test and training data and run many methods with many thresholds over each training and test set. The justification for this approach is that each method and threshold deals with sparsifying the columns differently and the most appropriate method for the data will select the most similar reduction of the data across different random sections of rows. We seek to minimize the absolute spectral norm distance between covariance matrix estimates from test and train data while retaining an intuitive threshold or number of factors.

Table 3 shows results of running the POET method with the soft 0.5 threshold recommended by the method authors using from three to seven factors and training data consisting of 0.5, 0.6, 0.7, 0.8 proportion of rows. The values displayed are the absolute spectral norm distance between covariance matrix estimates from test and train data of various proportions.

Training proportion	POET3 (soft)	POET4 (soft)	POET5 (soft)	POET6 (soft)	POET7 (soft)
0.5	799.46	799.38	798.92	799.29	799.41
0.6	21.16	21.08	20.66	20.68	20.38
0.6 (alternate seed)	2261.18	2260.68	2260.87	2260.86	2261.39
0.7	1114.49	1113.27	1114.14	1113.99	1114.027
0.8	1697.49	1696.21	1697.28	1696.57	1696.46

Table 3 POET method with soft 0.5 threshold over various training proportions and cluster numbers.

Table 3 reveals that the covariance matrix estimates vary greatly over the training proportion. There is a very strong variation in absolute spectral norm distance when the training proportion changes or the seed changes. We would expect to see more variation as the number of clusters changes but the spectral norm distance is relatively invariant with an increase in clusters. The strong variation in spectral norm distance is based entirely on the different rows used. This is exemplified most by seeing that for a training proportion of 0.6 using a different seed for the sampling creates huge differences in the spectral norm distance. These results reveal that there are immense problems with sparse covariance estimation when the data consists of a large number of dependent covariates.

Table 4 shows the results of running hard, soft, and adaptive threshold methods for various thresholds. The thresholds are chosen using the heuristics provided in [9] for hard and adaptive thresholds and [15] for soft thresholds. The heuristics suggest for all methods a threshold centered around 75 with slight variation. To study variation in spectral norm distance as the threshold changes we use nine thresholds ranging from 20 to 260. The results from the POET method reveal that the covariance estimation is very sensitive to the data used therefore we run the hard, soft, and adaptive threshold methods for 20 training sets each chosen to contain 0.5 proportion of rows. We present the average and standard deviation of spectral norm distances between these 20 training covariance estimates and 20 test covariance estimates. We expect methods that perform better to have lower spectral norm distances. Due to the slow computation of the POET method we were unable to perform this randomized averaging in the previous table.

method/thr	**20**	**50**	**80**	**110**	**140**	**170**	**200**	**230**	**260**
hard (mean)	1357	1310	1224	1153	1039	960	935	839	835
hard (sd)	1087	1068	1030	977	910	843	824	755	732
soft (mean)	1195	999	860	751	668	609	557	512	474
soft (sd)	987	860	756	672	603	546	498	455	421
adapt (mean)	109	109	109	109	109	109	109	109	109
adapt (sd)	84	84	84	84	84	84	84	84	84

Table 4 Mean and SD spectral norm distance for matrices estimated from 20 random train-test pairs over various thresholds suggested by heuristics.

Table 4 shows three relevant findings. First, the adaptive thresholding heuristic is not useful here since there is no change in the spectral norm distances as the threshold changes. Perhaps a much larger or much smaller threshold is needed for succesful adaptive threshold covariance estimation. Second, there is a general reduction in mean spectral norm distance for the hard- and soft- thresholding methods as the threshold increases. This suggests that the threshold selection heuristics are not useful here since a much larger threshold than suggested by the heuristic seems to yield a more suitable method. Third, the very large standard deviations of spectral norm distance across the twenty random train-test pairs suggests that we cannot have any confidence in the magnitude of the spectral norm distance.

Though there is no outstanding method among the four compared methods for this data the sparse estimates are still significantly different from the original singular empirical covariance matrix. The Cai [13] and Srivastara [14] hypothesis tests for matrix equality of selected estimated covariance matrices against original empirical covariance matrix find that the sparse covariance estimate is highly significantly different from the original empirical covariance matrix.

The comparison of these four methods for sparse covariance estimation of financial time series illustrates the immense difficulty of such covariance estimation. The estimation is extremely sensitive to data used and the choice of threshold and in general the threshold heuristics suggested still lead to very nonrobust estimates. A possibility for the inconclusivity of these results is that financial time series are highly dependent and therefore sparse methods are unable to decide on a common set of column reductions among subsamples of data from a shared dataset. Low-valued and robust spectral norm distances from a single method across various subsamples of data would imply that this method is dealing with the sparsifying data similarly across these subsamples and therefore that this method is strongly suited for the data. Further experimentation with thresholding and data preparation is required to ensure sufficient robustness of spectral norm distances for confidence in accurate sparse covariance estimation.

References

[1] Fan, J. Liao Y., Liu, H. (2015). An overview on the estimation of large covariance and precision matrices.

[2] Bickel, P., Levina, E. (2008). Regularized estimation of large covariance matrices. *The Annals of Statistics*, 36.1 199-227.

[3] Pun, Chi Seng (2018), "Low- and High-Dimensional Asset Prices Data", Mendeley Data, v3http://dx.doi.org/10.17632/ndxfrshm74.3

[4] Städler, N., Bühlmann, P. Missing values: sparse inverse covariance estimation and an extension to sparse regression. *Stat Comput* **22,** 219–235 (2012).

[5] Hui Zou, Trevor Hastie & Robert Tibshirani (2006) Sparse Principal Component Analysis, Journal of Computational and Graphical Statistics, 15:2, 265-286,

[6] Korkmaz S, Goksuluk D, Zararsiz G. MVN: An R Package for Assessing Multivariate Normality. The R Journal. 2014 6(2):151-162.

[7] Kyoungjae Lee and Kisung You (2019). CovTools: Statistical Tools for Covariance Analysis. R package version 0.5.3.

[8] Bickel PJ, Levina E (2008). "Covariance regularization by thresholding." The Annals of Statistics, 36(6), 2577–2604.

[9] Cai T, Liu W (2011). "Adaptive Thresholding for Sparse Covariance Matrix Estimation." Journal of the American Statistical Association, 106(494), 672–684.

[10] Fan J, Liao Y, Mincheva M (2013). "Large covariance estimation by thresholding principal orthogonal complements." Journal of the Royal Statistical Society: Series B (Statistical Methodology), 75(4), 603–680.

[11] Qi H, Sun D (2006). “A Quadratically Convergent Newton Method for Computing the Nearest Correlation Matrix.” SIAM Journal on Matrix Analysis and Applications, 28(2), 360–385.

[12] Fan, Liao and Mincheva (2012) "Large Covariance Estimation in Approximate Factor Models by Thresholding Principal Orthogonal Complements", manuscript of Princeton University, arXiv: 1201.0175

[13] Cai TT, Ma Z (2013). “Optimal hypothesis testing for high dimensional covariance matrices.” Bernoulli, 19(5B), 2359–2388.

[14] Srivastava MS, Yanagihara H, Kubokawa T (2014). “Tests for covariance matrices in high dimension with less sample size.” Journal of Multivariate Analysis, 130, 289–309.

[15] Antoniadias, A., Fan J. (2001). Regularization of Wavelet Approximations. Journal of the American Statistical Association: Vol. 96.

[16] Bien, J., Tibsharani, R. (2010) Sparse Estimation of a Covariance Matrix. Biometrika, 0.0 1-24.

R Code

```
### Siddarth Viswanathan
## STAT 8102 project 3


rm(list=ls())

library('ggplot2')
library('CovTools')
library('reshape2')
library('dtwclust')
library('sparsepca')
library('matrixcalc')
library('vcvComp')
library('POET')
library('Jmisc')


########
## 0. load data
########

setwd('/Users/siddarthviswanathan/Desktop/')

dat <- read.csv('russell2000_data.csv', stringsAsFactors = F)
dat[dat == '#N/A'] <- NA

ts_dat <- dat
ts_dat[,1] <- as.Date(ts_dat[,1]) # set date value

# 504 x 1550
short_ts_dat <- ts_dat[,as.numeric(sapply(ts_dat, function(x) sum(is.na(x))))==0]

final_mat <- as.matrix(short_ts_dat[,-1])
```

```
final_mat_df <- as.data.frame(final_mat)

test <- sapply(final_mat_df, function(x) mean(x))

short_ts_dat2 <- final_mat_df[,names(sort(test, decreasing=T)[-c(1:3)])]
short_ts_dat3 <- final_mat_df[,names(sort(test, decreasing=T)[-c(1:20)])]


######
## 1. Summarize the distribution and dependence structure of those predictors;
####

### 1a. hierarchical cormat

# understand hclust function
cormat <- round(cor(final_mat),3)

# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
  return(cormat)
}

reorder_cormat <- function(cormat){
  # Use correlation between variables as distance
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <-cormat[hc$order, hc$order]
}

# Reorder the correlation matrix
cormat <- reorder_cormat(cormat)
upper_tri <- get_upper_tri(cormat)
melted_cormat <- melt(upper_tri, na.rm = TRUE)

ggplot(data=melted_cormat, aes(x=Var1, y=Var2, fill=value))+geom_tile()

### 1b. spca

spca_1 <- spca(short_ts_dat3, k=5)
summary(spca_1)

spca_2 <- robspca(short_ts_dat3, k=10)
summary(spca_2)


### 1c. plot many lines
meltdf <- melt(short_ts_dat, id="Date")

p <- ggplot(meltdf, aes(x=Date, y=value, group=variable)) +
  theme_bw() +
  theme(panel.grid=element_blank()) +
  geom_line(size=0.2, alpha=0.4)




#######
##2. create and compare the covariance matrices
#######
covar_est_dat <- as.matrix(short_ts_dat2)

est1 <- cov(covar_est_dat, y=covar_est_dat, use="all.obs")



CovTest1.2013Cai(covar_est_dat, Sigma0=est2)

# reject=True, statistic 542.98, threshold 10.1
CovTest1.2013Cai(covar_est_dat, Sigma0=est5)
```

```
# reject = True, Statistic = 87.84, threshold = 1.64
CovTest1.2014Srivastava(covar_est_dat, Sigma0=est5)

CovTest1.2014Srivastava(covar_est_dat, Sigma0=est3)

# is the data mvn using many tests?
mvn(short_ts_dat2, mvnTest='mardia')[1] # no
mvn(short_ts_dat2, mvnTest='hz')[1] # no
mvn(short_ts_dat2, mvnTest='royston')[1] # no

#########
## 3. Which covariance matrix would you choose?
########
# get initial properties of original covariance matrix for intuition
median(as.numeric(diag(est1))) # 8.54
var(as.numeric(diag(est1)))^.5 # 194
mean(as.numeric(diag(est1))) #38.92

# run many tests
set.seed(310)
prop <- .65
n <- ceiling(prop*nrow(short_ts_dat2))
sample_indices <- sample(nrow(short_ts_dat2), n)

train_dat_50 <- short_ts_dat2[sample_indices,]
test_dat_50 <- short_ts_dat2[-sample_indices,]

train_dat_65 <- short_ts_dat2[sample_indices,]
test_dat_65 <- short_ts_dat2[-sample_indices,]

train_dat_80 <- short_ts_dat2[sample_indices,]
test_dat_80 <- short_ts_dat2[-sample_indices,]

train_dat <- as.matrix(train_dat_80)
test_dat <- as.matrix(test_dat_80)

thr_num <- .01

thr_vals <- c(.01, .05, .25, .5, .75, 1, 2, 5, 10, 25, 50, 75, 100, 150, 200)

spec_hard <- rep(NA, length(thr_vals))
spec_adap <- rep(NA, length(thr_vals))
spec_soft <- rep(NA, length(thr_vals))

for(k in 1:length(thr_vals)){
  print(k)
  thr_num <- thr_vals[k]

  train <- CovEst.hard(train_dat, thr=thr_num)$S
  test <- CovEst.hard(test_dat, thr=thr_num)$S
  spec_hard[k] <- abs(norm(train, type='2') - norm(test, type='2'))

  train <- CovEst.soft(train_dat, thr=thr_num)$S
  test <- CovEst.soft(test_dat, thr=thr_num)$S
  spec_soft[k] <- abs(norm(train, type='2') - norm(test, type='2'))

  train <- CovEst.adaptive(train_dat, thr=thr_num)$S
  test <- CovEst.adaptive(test_dat, thr=thr_num)$S
  spec_adap[k] <- abs(norm(train, type='2') - norm(test, type='2'))
}



# find optimal thresholding for adaptive
2*(log(1546)/504)^.5*119 # mean: 28-78

'''
short_dat <- as.matrix(short_ts_dat2)
num_cov <- 500
```

```
theta_ijs <- matrix(NA, nrow=num_cov, ncol=num_cov)
for(i in 1:num_cov){
  for(j in 1:num_cov){
    mean_i <- mean(short_dat[,i])
    mean_j <- mean(short_dat[,j])
    sigma_hat_ij <- est1[i,j]
    tot_sum <- 0
    for(k in 1:num_cov){
      tot_sum <- tot_sum + ((short_dat[k,i] - mean_i)*(short_dat[k,j] - mean_j) - sigma_hat_ij)^2
    }
    theta_ijs[i,j] <- tot_sum/num_cov
  }
}

mean(theta_ijs, na.rm=T)^.5
'''

start_time <- proc.time()
seeds <- sample(1000, 20)
thr_vals <- c(20, 50, 80, 110, 140, 170, 200, 230, 260)

results_mat_hard <- matrix(NA, nrow=length(seeds), ncol=length(thr_vals))
results_mat_soft <- matrix(NA, nrow=length(seeds), ncol=length(thr_vals))
results_mat_adap <- matrix(NA, nrow=length(seeds), ncol=length(thr_vals))

for(j in 1:length(seeds)){
  set.seed(seeds[j])
  prop <- .5
  n <- ceiling(prop*nrow(new_short_ts_dat2))
  sample_indices <- sample(nrow(new_short_ts_dat2), n)

  train_dat_50 <- new_short_ts_dat2[sample_indices,]
  test_dat_50 <- new_short_ts_dat2[-sample_indices,]

  train_dat <- as.matrix(train_dat_50)
  test_dat <- as.matrix(test_dat_50)

  for(k in 1:length(thr_vals)){
    thr_num <- thr_vals[k]

    train <- CovEst.hard(train_dat, thr=thr_num)$S
    test <- CovEst.hard(test_dat, thr=thr_num)$S
    results_mat_hard[j,k] <- abs(norm(train, type='2') - norm(test, type='2'))

    train <- CovEst.soft(train_dat, thr=thr_num)$S
    test <- CovEst.soft(test_dat, thr=thr_num)$S
    results_mat_soft[j,k] <- abs(norm(train, type='2') - norm(test, type='2'))

    train <- CovEst.adaptive(train_dat, thr=thr_num)$S
    test <- CovEst.adaptive(test_dat, thr=thr_num)$S
    results_mat_adap[j,k] <- abs(norm(train, type='2') - norm(test, type='2'))
  }

  print(j)
}
```