Gaussian framework_Simulation_Exploring smoothing effect

The hierarchical structure considered in this simulation study consists two levels where level one consists 2 series A and B, and the bottom level consists 4 series AA, AB, BA and BB. To make the aggregate levels smother than disaggregate levels, noise components were added to the aggregate levels as it was done in MinT paper. Gaussian probabilistic forecasts were obtained by reconciling means and variance covariance matrices with different reconciliation mathods, namely MinT shrink, MinT sample, OLS and also bottom-up method. Energy scores and Log scores were caclulated to evaluate these probabilistic forecasts.

```
#Let m be the number of bottom level series
#Let N be the size of the series
N<-510
L<-500
m<-4
B<-1000
k<-10000

#Randomly choosing AR and MA parameters to generate data
AR_coef<-runif(n=m, min = 0.4, max = 0.7)
MA_coef<-runif(n=m, min = 0.4, max = 0.7)

#Generating bottom level covariance matrix
#Bottom_pop_cov<-symMatrix(c(4,3.6,5,2.7,3.5,5,1.7,1.5,0.9,3,2.2,3,1.9,3.2,
#                            6,0.8,1.6,0.7,1.4,2.7,4,-1.4,-0.6,-1.3,1.7,
#                            3.1,4,8),m,m, upper = TRUE)

Bottom_pop_cov<-diag(2, m, m)


E<-list()

#Initializing the variables.
#P_BU : I_m
#P_OLS :  (S'S)^(-1)S'
#P_MinT.shr : (S'Inv_W S)^(-1)S'Inv_W
#P_MinT.sam : (S'Inv_sig.sam S)^(-1)S'nv_sig.sam


ES_Bottum.up<-numeric(0)
ES_OLS<-numeric(0)
ES_MinT.shr<-numeric(0)
ES_MinT.sam<-numeric(0)
ES_unreconciled<-numeric(0)

LS_Bottum.up<-numeric(0)
LS_OLS<-numeric(0)
LS_MinT.shr<-numeric(0)
LS_MinT.sam<-numeric(0)
```

```
for(j in  1:B)
{

  #Randomly generating errors
  E[[j]]<-mvrnorm(n = N, mu = rep(0, m), Sigma = Bottom_pop_cov)

  #Generating the bottom level series. Each series were generated from
  #ARMA(1,1) model where the parameters were randomly selected from the
  #defined parameter space
  Bottom_level<-matrix(0, nrow = N, ncol = m)

  for(i in 1:m)
  {
    Bottom_level[,i]<-arima.sim(list(order=c(1,0,1),ar=AR_coef[i],
                                     ma=MA_coef[i]), n = N, innov = E[[j]][,i])
  }

  Vt<-rnorm(n = N, mean = 0, sd = sqrt(10))
  Wt<-rnorm(n = N, mean = 0, sd = sqrt(7))

  Bottom_level_noisy<-matrix(0, nrow = N, ncol = m)

  Bottom_level_noisy[,1]<-Bottom_level[,1]+Vt-0.5*Wt
  Bottom_level_noisy[,2]<-Bottom_level[,2]-Vt-0.5*Wt
  Bottom_level_noisy[,3]<-Bottom_level[,3]+Vt+0.5*Wt
  Bottom_level_noisy[,4]<-Bottom_level[,4]-Vt+0.5*Wt


#Generating the hierarchy

  Hierarchy<-suppressMessages(hts(Bottom_level_noisy, list(2, c(2,2))))
  AllTS<-allts(Hierarchy)
  n<-ncol(AllTS)

#Generating the summing matrix
  S<-smatrix(Hierarchy)

  Training <- AllTS[1:L,]
  Testing <- AllTS[(L+1):N,]

#Model fitting, forecasting and obtaining residuals

  h<-1
  Residuals_all<-matrix(NA, nrow = nrow(Training), ncol = n)
  Base_forecasts<-matrix(NA, nrow = h, ncol = n)

  for(i in 1:n)
  {
    fit<-auto.arima(Training[,i])
    Base_forecasts[,i]<-forecast(fit, h=1)$mean
    Residuals_all[,i]<-Training[,i]-fitted(fit)
  }
```

```r
#Calculating in-sample variance covariance matrix
n1<-nrow(Residuals_all)
Sigma_sample<-crossprod(Residuals_all)/n1


#Obtaining shrinkage estimator for var-cov matrix of in-sample errors
#(From MinT package)

lowerD <- function(x)
{
  n2 <- nrow(x)
  return(diag(apply(x, 2, crossprod) / n2))
}

shrink.estim <- function(x, tar)
{
  if (is.matrix(x) == TRUE && is.numeric(x) == FALSE)
    stop("The data matrix must be numeric!")
  p <- ncol(x)
  n2 <- nrow(x)
  covm <- crossprod(x) / n2
  corm <- cov2cor(covm)
  xs <- scale(x, center = FALSE, scale = sqrt(diag(covm)))
  v <- (1/(n2 * (n2 - 1))) * (crossprod(xs^2) - 1/n2 * (crossprod(xs))^2)
  diag(v) <- 0
  corapn <- cov2cor(tar)
  d <- (corm - corapn)^2
  lambda <- sum(v)/sum(d)
  lambda <- max(min(lambda, 1), 0)
  shrink.cov <- lambda * tar + (1 - lambda) * covm
  return(list(shrink.cov, c("The shrinkage intensity lambda is:",
                            round(lambda, digits = 4))))
}

targ<-lowerD(Residuals_all)
shrink<-shrink.estim(Residuals_all,targ)
W.h<-shrink[[1]]
Unreconsiled_shrinkage_cov_mat<-W.h

n1<-nrow(Residuals_all)
Sigma_sample<-crossprod(Residuals_all)/n1

Inv_W.h<-solve(W.h)
Inv_sig.sam<-solve(Sigma_sample)

#Calculating different P matrices
Null.ma<-matrix(0,m,(n-m))
P_BU<-cbind(Null.ma, diag(1,m,m))
P_OLS<-solve(t(S)%*%S)%*%t(S)
P_MinT.shr<-solve(t(S)%*%Inv_W.h%*%S)%*%t(S)%*%Inv_W.h
P_MinT.sam<-solve(t(S)%*%Inv_sig.sam%*%S)%*%t(S)%*%Inv_sig.sam

#Obtainig reconciled point forecasts with different reconciliation
```

```r
#methods
Reconciled_point.forecasts_BU<-S%*%P_BU%*%t(Base_forecasts)

Reconciled_point.forecasts_Mint.shr<-MinT(Base_forecasts, nodes = Hierarchy$nodes,
                                          residual = Residuals_all, covariance = "shr",
                                          algorithms = "lu", keep = "all")
Reconciled_point.forecasts_Mint.sam<-MinT(Base_forecasts, nodes = Hierarchy$nodes,
                                          residual = Residuals_all, covariance = "sam",
                                          algorithms = "lu", keep = "all")
Reconciled_point.forecasts_OLS<-combinef(Base_forecasts, Hierarchy$nodes, weights = NULL,
                                         algorithms = "lu", keep = "all" )

#Obtainig reconciled variance covariance matrices with different reconciliation
#methods

Sigma.tilde_BU<-S%*%P_BU%*%W.h%*%t(P_BU)%*%t(S)
Sigma.tilde_OLS<-S%*%P_OLS%*%W.h%*%t(P_OLS)%*%t(S)
Sigma.tilde_MinT.shr<-S%*%P_MinT.shr%*%W.h%*%t(P_MinT.shr)%*%t(S)
Sigma.tilde_MinT.sam<-S%*%P_MinT.sam%*%Sigma_sample%*%t(P_MinT.sam)%*%t(S)


#Evaluating the density forecasts using Energy score

#Calculating Enery score to evaluate

Energy_score<-function(Data)
{
  ES_1_eval<-numeric(0)
  ES_2_eval<-numeric(0)

  d1_eval<-(Data)-
    matrix(rep(Testing[1,],k),k,n,byrow = TRUE)
  ES_1_eval<-apply(d1_eval, 1, function(x) sqrt(sum(x^2)))

  d2_eval<-(Data)[1:k-1,]-(Data)[2:k,]
  ES_2_eval<-apply(d2_eval, 1, function(x) sqrt(sum(x^2)))
  ES_eval<-mean(ES_1_eval)-mean(ES_2_eval)/2

  return(ES_eval)

}

#Obtaining a random variable from the possible forecast Gaussian densities

X_BU<-mvrnorm(n=k, mu=Reconciled_point.forecasts_BU,
              Sigma = Sigma.tilde_BU)

X_OLS<-mvrnorm(n=k, mu=Reconciled_point.forecasts_OLS, Sigma = Sigma.tilde_OLS)

X_MinT.shr<-mvrnorm(n=k, mu=Reconciled_point.forecasts_Mint.shr,
                    Sigma = Sigma.tilde_MinT.shr)

X_MinT.sam<-mvrnorm(n=k, mu=Reconciled_point.forecasts_Mint.sam,
```

```
                    Sigma = Sigma.tilde_MinT.sam)

  X_unreconciled<-mvrnorm(n=k, mu=Base_forecasts,
                          Sigma = Unreconsiled_shrinkage_cov_mat)



  #Calculating Energy score for predicive densities

  ES_Bottum.up[j] <-Energy_score(X_BU)
  ES_OLS[j] <-Energy_score(X_OLS)
  ES_MinT.shr[j] <-Energy_score(X_MinT.shr)
  ES_MinT.sam[j] <-Energy_score(X_MinT.sam)
  ES_unreconciled[j] <-Energy_score(X_unreconciled)


  #Evaluating the density forecasts using Log score

  Eigen_Reconciled_sigma_MinT.shr<-zapsmall(eigen(Sigma.tilde_MinT.shr)$values)
  Eigen_Reconciled_sigma_MinT.sam<-zapsmall(eigen(Sigma.tilde_MinT.sam)$values)
  Eigen_Reconciled_sigma_OLS<-zapsmall(eigen(Sigma.tilde_OLS)$values)
  Eigen_Reconciled_sigma_BU<-zapsmall(eigen(Sigma.tilde_BU)$values)

  G.Inv_Reconciled_sigma_MinT.shr<-ginv(Sigma.tilde_MinT.shr)
  G.Inv_Reconciled_sigma_MinT.sam<-ginv(Sigma.tilde_MinT.sam)
  G.Inv_Reconciled_sigma_OLS<-ginv(Sigma.tilde_OLS)
  G.Inv_Reconciled_sigma_BU<-ginv(Sigma.tilde_BU)

  Mean_shift_MinT.shr<-c(AllTS[n+1,]-Reconciled_point.forecasts_Mint.shr)
  Mean_shift_MinT.sam<-c(AllTS[n+1,]-Reconciled_point.forecasts_Mint.sam)
  Mean_shift_OLS<-c(AllTS[n+1,]-Reconciled_point.forecasts_OLS)
  Mean_shift_BU<-c(AllTS[n+1,]-Reconciled_point.forecasts_BU)


  LS_MinT.shr[j]<-(1/2)*(log(prod(Eigen_Reconciled_sigma_MinT.shr[Eigen_Reconciled_sigma_MinT.shr!=0]))
                                    +t(Mean_shift_MinT.shr)%*%G.Inv_Reconciled_sigma_MinT.shr%*%Mean_

  LS_MinT.sam[j]<-(1/2)*(log(prod(Eigen_Reconciled_sigma_MinT.sam[Eigen_Reconciled_sigma_MinT.sam!=0]))
                                    +t(Mean_shift_MinT.sam)%*%G.Inv_Reconciled_sigma_MinT.sam%*%Mean_

  LS_OLS[j]<-(1/2)*(log(prod(Eigen_Reconciled_sigma_OLS[Eigen_Reconciled_sigma_OLS!=0]))
                                +t(Mean_shift_OLS)%*%G.Inv_Reconciled_sigma_OLS%*%Mean_shift_OLS)

  LS_Bottum.up[j]<-(1/2)*(log(prod(Eigen_Reconciled_sigma_BU[Eigen_Reconciled_sigma_BU!=0]))
                            +t(Mean_shift_BU)%*%G.Inv_Reconciled_sigma_BU%*%Mean_shift_BU)


}



Mean_ES_Bottum.up<-round(mean(ES_Bottum.up), digits = 4)
Mean_ES_OLS<-round(mean(ES_OLS), digits = 4)
```

```r
Mean_ES_MinT.shr<-round(mean(ES_MinT.shr), digits = 4)
Mean_ES_MinT.sam<-round(mean(ES_MinT.sam), digits = 4)
Mean_ES_unreconciled<-round(mean(ES_unreconciled), digits = 4)

Mean_LS_Bottum.up<-round(mean(LS_Bottum.up), digits = 4)
Mean_LS_OLS<-round(mean(LS_OLS), digits = 4)
Mean_LS_MinT.shr<-round(mean(LS_MinT.shr), digits = 4)
Mean_LS_MinT.sam<-round(mean(LS_MinT.sam), digits = 4)



Method<-c("Sigma_unreconciled", "Sigma.tilde_Bottom.up", "Sigma.tilde_OLS", "Sigma.tilde_MinT.sam",
          "Sigma.tilde_MinT.shr")
Mean_Energy_score<-c(Mean_ES_unreconciled, Mean_ES_Bottum.up, Mean_ES_OLS,
                     Mean_ES_MinT.sam, Mean_ES_MinT.shr)
Mean_Log_score<-c("-", Mean_LS_Bottum.up, Mean_LS_OLS, Mean_LS_MinT.sam, Mean_LS_MinT.shr)
Eval_Prob_forecasts <- data.frame(Method, Mean_Energy_score, Mean_Log_score)

kable(Eval_Prob_forecasts, format = "markdown", align = "c")
```

Method	Mean_Energy_score	Mean_Log_score
Sigma_unreconciled	6.6334	-
Sigma.tilde_Bottom.up	6.9428	8.8095
Sigma.tilde_OLS	6.4987	9.6556
Sigma.tilde_MinT.sam	6.4305	10.1607
Sigma.tilde_MinT.shr	6.4291	10.1235

```r
#Testing Unreconciled vs reconciled prob.forecasts

#Unreconciled Vs Bottom up
Sigma_unrecon.vs.BU<-mean((ES_unreconciled - ES_Bottum.up)^2)
t_unrecon.vs.BU<-sqrt(B)*(Mean_ES_unreconciled - Mean_ES_Bottum.up)/sqrt(Sigma_unrecon.vs.BU)
p.val_unrecon.vs.BU<-2*pnorm(-abs(t_unrecon.vs.BU))

#Unreconciled Vs OLS
Sigma_unrecon.vs.OLS<-mean((ES_unreconciled - ES_OLS)^2)
t_unrecon.vs.OLS<-sqrt(B)*(Mean_ES_unreconciled - Mean_ES_OLS)/sqrt(Sigma_unrecon.vs.OLS)
p.val_unrecon.vs.OLS<-2*pnorm(-abs(t_unrecon.vs.OLS))

#Unreconciled Vs Mint.shr
Sigma_unrecon.vs.MinT.shr<-mean((ES_unreconciled - ES_MinT.shr)^2)
t_unrecon.vs.MinT.shr<-sqrt(B)*(Mean_ES_unreconciled - Mean_ES_MinT.shr)/sqrt(Sigma_unrecon.vs.MinT.shr
p.val_unrecon.vs.MinT.shr<-2*pnorm(-abs(t_unrecon.vs.MinT.shr))

#Unreconciled Vs Mint.sam
Sigma_unrecon.vs.MinT.sam<-mean((ES_unreconciled - ES_MinT.sam)^2)
t_unrecon.vs.MinT.sam<-sqrt(B)*(Mean_ES_unreconciled - Mean_ES_MinT.sam)/sqrt(Sigma_unrecon.vs.MinT.sam
p.val_unrecon.vs.MinT.sam<-2*pnorm(-abs(t_unrecon.vs.MinT.sam))

Unreconciled_vs<-c("Bottom_up", "OLS", "MinT.sam",
                   "MinT.shr")
Test_statistic<-c(t_unrecon.vs.BU, t_unrecon.vs.OLS, t_unrecon.vs.MinT.sam,
```

```
                  t_unrecon.vs.MinT.shr)
p_value<-c(p.val_unrecon.vs.BU, p.val_unrecon.vs.OLS, p.val_unrecon.vs.MinT.sam,
           p.val_unrecon.vs.MinT.shr)
Unreconciled_vs_reconciled<-data.frame(Unreconciled_vs,Test_statistic,
                                       p_value)



#Bottom up Vs OLS wrt ES
Sigma_BU.vs.OLS<-mean((ES_Bottum.up - ES_OLS)^2)
t_BU.vs.OLS<-sqrt(B)*(Mean_ES_Bottum.up - Mean_ES_OLS)/sqrt(Sigma_BU.vs.OLS)
p.val_BU.vs.OLS<-2*pnorm(-abs(t_BU.vs.OLS))

#Bottom up Vs MinT.sam wrt ES
Sigma_BU.vs.MinT.sam<-mean((ES_Bottum.up - ES_MinT.sam)^2)
t_BU.vs.MinT.sam<-sqrt(B)*(Mean_ES_Bottum.up - Mean_ES_MinT.sam)/sqrt(Sigma_BU.vs.MinT.sam)
p.val_BU.vs.MinT.sam<-2*pnorm(-abs(t_BU.vs.MinT.sam))

#Bottom up Vs MinT.shr wrt ES
Sigma_BU.vs.MinT.shr<-mean((ES_Bottum.up - ES_MinT.shr)^2)
t_BU.vs.MinT.shr<-sqrt(B)*(Mean_ES_Bottum.up - Mean_ES_MinT.shr)/sqrt(Sigma_BU.vs.MinT.shr)
p.val_BU.vs.MinT.shr<-2*pnorm(-abs(t_BU.vs.MinT.shr))

#OLS Vs MinT.sam wrt ES
Sigma_OLS.vs.MinT.sam<-mean((ES_OLS - ES_MinT.sam)^2)
t_OLS.vs.MinT.sam<-sqrt(B)*(Mean_ES_OLS - Mean_ES_MinT.sam)/sqrt(Sigma_OLS.vs.MinT.sam)
p.val_OLS.vs.MinT.sam<-2*pnorm(-abs(t_OLS.vs.MinT.sam))

# MinT.shr Vs MinT.sam wrt ES
Sigma_MinT.shr.vs.MinT.sam<-mean((ES_MinT.shr - ES_MinT.sam)^2)
t_MinT.shr.vs.MinT.sam<-sqrt(B)*(Mean_ES_MinT.shr - Mean_ES_MinT.sam)/sqrt(Sigma_MinT.shr.vs.MinT.sam)
p.val_MinT.shr.vs.MinT.sam<-2*pnorm(-abs(t_MinT.shr.vs.MinT.sam))


#OLS Vs MinT.shr wrt ES
Sigma_OLS.vs.MinT.shr<-mean((ES_OLS - ES_MinT.shr)^2)
t_OLS.vs.MinT.shr<-sqrt(B)*(Mean_ES_OLS - Mean_ES_MinT.shr)/sqrt(Sigma_OLS.vs.MinT.shr)
p.val_OLS.vs.MinT.shr<-2*pnorm(-abs(t_OLS.vs.MinT.shr))

#Bottom up Vs OLS wrt LS
Sigma_BU.vs.OLS_LS<-mean((LS_Bottum.up - LS_OLS)^2)
t_BU.vs.OLS_LS<-sqrt(B)*(Mean_LS_Bottum.up - Mean_LS_OLS)/sqrt(Sigma_BU.vs.OLS_LS)
p.val_BU.vs.OLS_LS<-2*pnorm(-abs(t_BU.vs.OLS_LS))

#Bottom up Vs MinT.sam wrt ES
Sigma_BU.vs.MinT.sam_LS<-mean((LS_Bottum.up - LS_MinT.sam)^2)
t_BU.vs.MinT.sam_LS<-sqrt(B)*(Mean_LS_Bottum.up - Mean_LS_MinT.sam)/sqrt(Sigma_BU.vs.MinT.sam_LS)
p.val_BU.vs.MinT.sam_LS<-2*pnorm(-abs(t_BU.vs.MinT.sam_LS))

#Bottom up Vs MinT.shr wrt LS
Sigma_BU.vs.MinT.shr_LS<-mean((LS_Bottum.up - LS_MinT.shr)^2)
t_BU.vs.MinT.shr_LS<-sqrt(B)*(Mean_LS_Bottum.up - Mean_LS_MinT.shr)/sqrt(Sigma_BU.vs.MinT.shr_LS)
p.val_BU.vs.MinT.shr_LS<-2*pnorm(-abs(t_BU.vs.MinT.shr_LS))
```

```r
#OLS Vs MinT.shr wrt LS
Sigma_OLS.vs.MinT.shr_LS<-mean((LS_OLS - LS_MinT.shr)^2)
t_OLS.vs.MinT.shr_LS<-sqrt(B)*(Mean_LS_OLS - Mean_LS_MinT.shr)/sqrt(Sigma_OLS.vs.MinT.shr_LS)
p.val_OLS.vs.MinT.shr_LS<-2*pnorm(-abs(t_OLS.vs.MinT.shr_LS))

#OLS Vs MinT.sam wrt LS
Sigma_OLS.vs.MinT.sam_LS<-mean((LS_OLS - LS_MinT.sam)^2)
t_OLS.vs.MinT.sam_LS<-sqrt(B)*(Mean_LS_OLS - Mean_LS_MinT.sam)/sqrt(Sigma_OLS.vs.MinT.sam_LS)
p.val_OLS.vs.MinT.sam_LS<-2*pnorm(-abs(t_OLS.vs.MinT.sam_LS))

# MinT.shr Vs MinT.sam wrt ES
Sigma_MinT.shr.vs.MinT.sam_LS<-mean((LS_MinT.shr - LS_MinT.sam)^2)
t_MinT.shr.vs.MinT.sam_LS<-sqrt(B)*(Mean_LS_MinT.shr - Mean_LS_MinT.sam)/sqrt(Sigma_MinT.shr.vs.MinT.sa
p.val_MinT.shr.vs.MinT.sam_LS<-2*pnorm(-abs(t_MinT.shr.vs.MinT.sam_LS))

Comparison<-c("MinT.shr Vs MinT.sam", "MinT.shr Vs OLS", "MinT.shr Vs BU",
              "MinT.sam Vs OLS", "MinT.sam Vs BU", "OLS Vs BU")
P_val_ES<-c(p.val_MinT.shr.vs.MinT.sam, p.val_OLS.vs.MinT.shr, p.val_BU.vs.MinT.shr,
            p.val_OLS.vs.MinT.sam, p.val_BU.vs.MinT.sam, p.val_BU.vs.OLS)
P_val_LS<-c(p.val_MinT.shr.vs.MinT.sam_LS, p.val_OLS.vs.MinT.shr_LS,
            p.val_BU.vs.MinT.shr_LS, p.val_OLS.vs.MinT.sam_LS, p.val_BU.vs.MinT.sam_LS,
            p.val_BU.vs.OLS_LS)
P_value<-data.frame(Comparison, P_val_ES, P_val_LS)

kable(Eval_Prob_forecasts, format = "markdown", align = "c")
```

Method	Mean_Energy_score	Mean_Log_score
Sigma_unreconciled	6.6334	-
Sigma.tilde_Bottom.up	6.9428	8.8095
Sigma.tilde_OLS	6.4987	9.6556
Sigma.tilde_MinT.sam	6.4305	10.1607
Sigma.tilde_MinT.shr	6.4291	10.1235

```r
kable(Unreconciled_vs_reconciled, format = "markdown", align = "c")
```

Unreconciled_vs	Test_statistic	p_value
Bottom_up	-11.20293	0
OLS	20.48817	0
MinT.sam	11.28998	0
MinT.shr	12.78189	0

```r
kable(P_value, format = "markdown", align = "c")
```

Comparison	P_val_ES	P_val_LS
MinT.shr Vs MinT.sam	0.7774479	0
MinT.shr Vs OLS	0.0000004	0
MinT.shr Vs BU	0.0000000	0
MinT.sam Vs OLS	0.0000225	0
MinT.sam Vs BU	0.0000000	0

Comparison	P__val__ES	P__val__LS
OLS Vs BU	0.0000000	0