Academic Journal of Applied Mathematical Sciences
ISSN(e): 2415-2188, ISSN(p): 2415-5225
Vol. 7, Issue. 4, pp: 187-191, 2021
URL: https://arpgweb.com/journal/journal/17
DOI: https://doi.org/10.32861/ajams.74.187.191

Academic Research Publishing Group

Original Research **Open Access**

The Modified Increment Method for Eigenspace Model

Chunjie Wei
School of Mathematics and Statistics, Shandong University of Technology, Zibo, Shandong, China

Jian Wang (Corresponding Author)
School of Mathematics and Statistics, Shandong University of Technology, Zibo, Shandong, China
Email: wjzhenhua@126.com

Article History
Received: 12 June, 2021
Revised: 23 July, 2021
Accepted: 9 August, 2021
Published: 13 August, 2021

Abstract

Eigenspace is a convenient way to represent sets of observations with widespread applications, so it is necessary to accurately calculate the eigenspace of data. With the advent of the era of big data, the increasing and updating of data brings great challenges to the solution of eigenspace. Hall, *et al.* [1], proposed that incremental method could update the eigenspace of data online, which reduces computational costs and storage space. In this paper, the updating coefficient of sample covariance matrix in incremental method is modified. Numerical analysis shows that the modified updating form has better performance.

Keywords: Eigenspace; Sample covariance matrix; Incremental method; Online update.

1. Introduction

Eigenspace is a convenient way to represent observation data and has a wide range of applications. Sharma, *et al.* [2], studied the application of eigenspace method in the detection and location of myocardial infarction. Li, *et al.* [3], studied the generalized sidelobe cancellation beamforming based on the eigenspace in medical ultrasound imaging. Ye, *et al.* [4], used eigenspace direct sum for cross-age face recognition. Many scholars also conduct research on large-scale eigenspace. Wen, *et al.* [5], proposed an unconstrained trajectory penalty minimization model, and established its equivalence with the eigenvalue problem. Ren, *et al.* [6], proposed an eigenspace divide-and-conquer method, which proved that the method had strong robustness and good expansibility for the dimension of the problem. The eigenspace can be calculated by performing eigenvalue decomposition (EVD) on the sample covariance matrix of the data. However, with the advent of the era of big data, the increase and continuous update of data brings great challenges to the solution of the eigenspace. When the data increases, adding the increased data to the original data to recalculate, not only the operation is complicated, but also the calculation and storage will be increased.

In 1998, Hall, *et al.* [1] described a construction method for progressively increasing the amount of observations in the eigenspace model, specifying the change in origin and the change in the number of eigenvectors required for the base set. This method updates the eigenspace online, and saves the cost of storage space and time without sacrificing the estimation accuracy. However, when the sample covariance matrix is updated, there is a certain error in the coefficient, which leads to a decrease in the estimation accuracy. This paper revises the updated sample covariance matrix in the incremental method, and the analysis of simulated data set and real data set shows that the revised updated form improves the estimation accuracy.

2. Eigenspace Update

2.1. Eigenspace Update with Incremental Method

Hall, *et al.* [1], proposed an incremental method to update the eigenspace, which can be used to update the results of previous calculations. This method calculates a smaller eigenspace model representing the observed values, which is more effective for classification. The observation data matrix $X = (\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}}, \cdots, \mathbf{x}_{\mathrm{N}}^{\mathrm{T}})^{\mathrm{T}}$ of p-dimension has N samples, where sample mean is $\bar{\mathbf{x}} = \sum_{i=1}^{\mathrm{N}} \mathbf{x}_i / \mathrm{N}$. The eigenvalue decomposition of the sample covariance matrix can be obtained

$$C = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^{\mathrm{T}} = U \Lambda U^{\mathrm{T}}, \tag{2.1}$$

where $U = (\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_p)$ is orthogonal eigenvector matrix, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_p)$ as the diagonal matrix, and satisfies $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq \mathbf{0}$. Construct the current eigenspace $\Omega = (\bar{\mathbf{x}}, U, \Lambda, \mathrm{N})$.

When the $(N+1)$th sample $\mathbf{y}$ comes, it's easy to get the updated mean can be written as

$$\bar{\mathbf{x}}' = \frac{1}{N+1}(N\bar{\mathbf{x}} + \mathbf{y}). \tag{2.2}$$

The updated covariance matrix is expressed as

$$C' = \frac{1}{N+1}C + \frac{N}{(N+1)^2}\mathbf{y}'(\mathbf{y}')^T, \tag{2.3}$$

where $\mathbf{y}' = \mathbf{y} - \bar{\mathbf{x}}$. The eigenvalue decomposition is performed for the updated sample covariance matrix $C' = U'\Lambda'U'^T$. Thus, we have the updated eigenspace $\Omega' = (\bar{\mathbf{x}}', U', \Lambda', N+1)$.

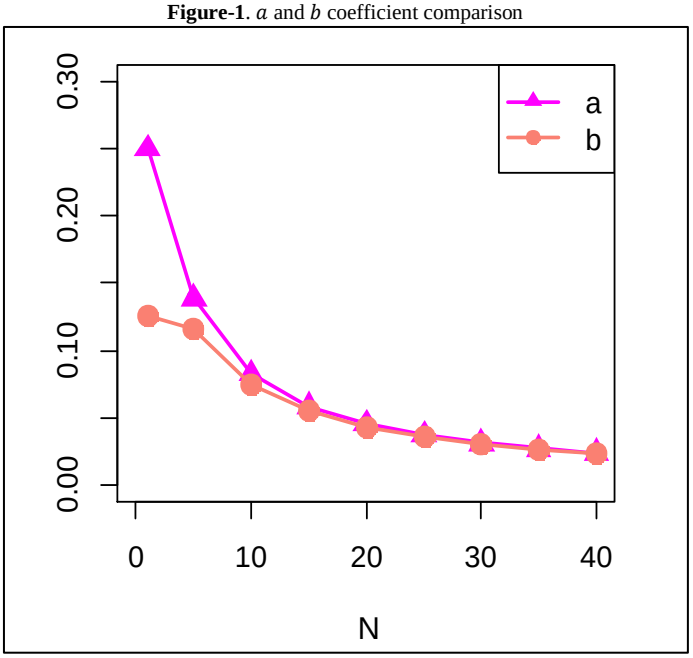
2.2. Eigenspace Update with Modified Incremental Method

We find that there are some errors in the update process above, and we correct the update of the sample covariance matrix. When the $(N+1)$th sample $\mathbf{y}$ arrives, it is easy to get the updated mean $\bar{\mathbf{x}}' = (N\bar{\mathbf{x}} + \mathbf{y})/(N+1)$. Derive from the definition of the sample covariance matrix, we can get the updated covariance matrix is expressed as

$$\begin{aligned}
C' &= \frac{1}{N+1}\sum_{i=1}^{N+1}(\mathbf{x}_i - \bar{\mathbf{x}}')(\mathbf{x}_i - \bar{\mathbf{x}}')^T \\
&= \frac{1}{N+1}\left[\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T + (\mathbf{y} - \bar{\mathbf{x}}')(\mathbf{y} - \bar{\mathbf{x}}')^T\right] \\
&= \frac{1}{N+1}C + \frac{1}{N+1}\left(\mathbf{y} - \frac{N\bar{\mathbf{x}} + \mathbf{y}}{N+1}\right)\left(\mathbf{y} - \frac{N\bar{\mathbf{x}} + \mathbf{y}}{N+1}\right)^T \\
&= \frac{1}{N+1}C + \frac{1}{N+1}\left(\frac{N\mathbf{y}}{N+1} - \frac{N\bar{\mathbf{x}}}{N+1}\right)\left(\frac{N\mathbf{y}}{N+1} - \frac{N\bar{\mathbf{x}}}{N+1}\right)^T \\
&= \frac{1}{N+1}C + \frac{N^2}{(N+1)^3}(\mathbf{y} - \bar{\mathbf{x}})(\mathbf{y} - \bar{\mathbf{x}})^T \\
&= \frac{1}{N+1}C + \frac{N^2}{(N+1)^3}\mathbf{y}'(\mathbf{y}')^T.
\end{aligned}$$

The result derived is $C' = \frac{1}{N+1}C + \frac{N}{(N+1)^2}\mathbf{y}'(\mathbf{y}')^T$ by Hall, et al, the result of our derivation is $C' = \frac{1}{N+1}C + \frac{N^2}{(N+1)^3}\mathbf{y}'(\mathbf{y}')^T$. We find that there is only a difference of $N/(N+1)$. For the convenience of representation, we record the coefficient deduced by Hall, *et al.* [1] as $a = \frac{N}{(N+1)^2}$, and our modified coefficient as $b = \frac{N^2}{(N+1)^3}$, with the sample change of the quantity and the change of two different coefficients are shown in Figure 1.

Figure-1. a and b coefficient comparison



As shown in Figure 1, when $N < 10$, the difference between a and b is very large. As the number of samples N increases, the difference between a and b becomes smaller and smaller. It shows that in the case of a small number of samples, the error of the coefficient is very large. If the inaccurate coefficient is substituted into the formula for updating, there will be a certain error every time it is updated, resulting in a continuous increase in the final error. In addition, the smaller the training sample N_0 is, the larger the coefficient difference will be, and the more updates will be. Because online updates are very dependent on the results of the previous update, this will lead to very inaccurate estimates, which will affect subsequent calculations. The accuracy of the update coefficient is very important in the online calculation.

Table-1. Comparison result of a and b when N <10

N	1	2	3	4	5	6	7	8	9
a	0.25	0.22	0.19	0.16	0.14	0.12	0.11	0.1	0.09
b	0.125	0.15	0.14	0.13	0.12	0.1	0.1	0.09	0.08

When $N < 10$, the values of a and b are shown in Table 1. It can be seen that the two coefficients are very different. When $N = 1$ and 2, a is basically twice b. As N increases, the difference is gradually reduced, but the minimum is 0.01. This is very large for the coefficient of formula updated, which may cause large errors in subsequent calculations.

3. Numerical Analysis

In this section, we use simulated data set and real data set to test the errors of the two update forms. Define Hall et al' s incremental update method as method 1, and our revised method as method 2. The mean square error (MSE) of the sample covariance matrix is used to represent the size of the error, and the mean square error is defined as follows

$$MSE(\hat{C}) = \frac{1}{p^2}\left\|\hat{C} - C\right\|_F^2,$$

where $\hat{C}$ is the sample covariance matrix obtained by incremental update, and C is the sample covariance matrix directly calculated off-line.

Simulation: Fixed $N = 16, p = 8$, we set $X \sim N_p(0, \Sigma)$, where $\sigma_{ij} = \sigma_{ji} = \sqrt{\sigma_{ii}}\sqrt{\sigma_{jj}} \cdot r_{ij}$, $|r_{ij}| \in [0.5,1]$, $i \neq j$. Generate a data matrix X, take $N/2$ sample data as training samples, and update the remaining $N/2$ incrementally. The mean square error values of the two methods to update the sample covariance matrix are shown in Table 2. The error is 1.654074 of modified update form, and the error of method 1 is 1.818826, indicating that the sample covariance matrix estimated by method 1 has more big error.

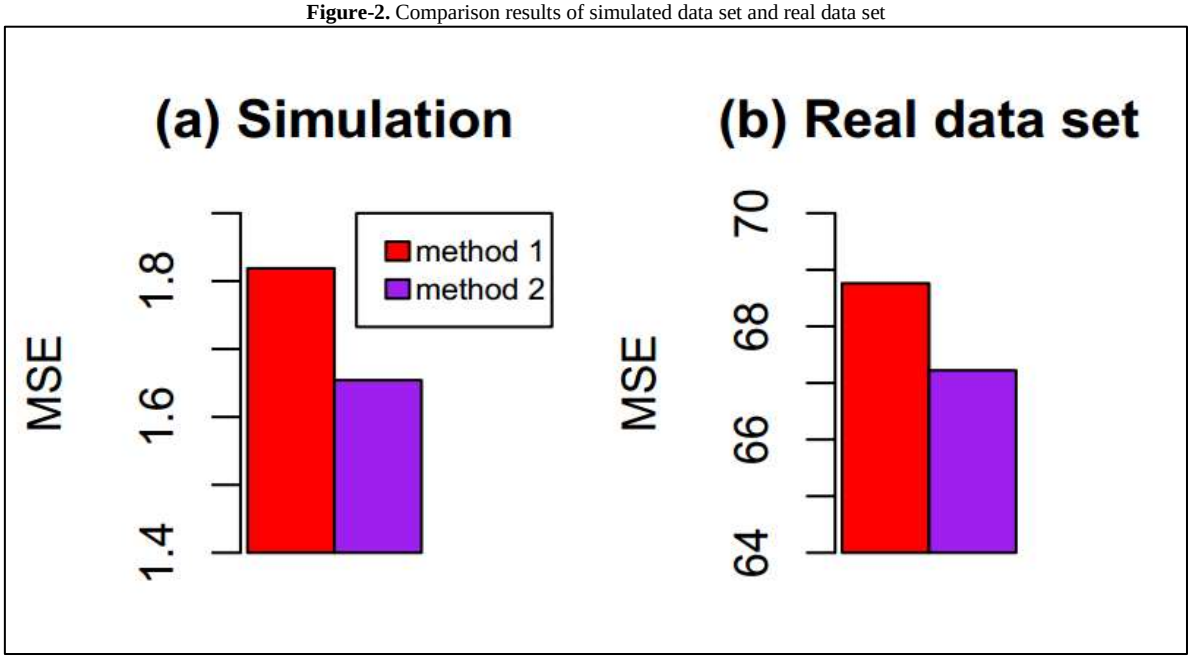
Real data set: The Immunotherapy data set is derived from the UCI database, and has 90 samples and 8 variables. As with the simulated data, $N/2$ sample data is taken as the training sample. As can be seen from Table 2, the error of the revised updated form is 67.22176, and the error of method 1 is 68.76202, which is consistent with the simulation data. It indicates that even if the two coefficients are very close to each other in the case of a larger sample, method 1 will still lead to a larger error, which indicates that it is very necessary to revise the update coefficient.

Table-2. Comparison results of simulated data set and real data set

method	simulated data set MSE	real data set MSE
method 1	1.818826	68.76202
method 2	1.654074	67.22176

We transformed the test results of the simulated data set and real data set into Figure 2. It is obvious that the test results of the simulated data are consistent with the real data. The MSE value of method 1 has larger error than method 2. The simulation data samples are 16 and the real data samples are 90. It shows that there are great errors in method 1 in the case of large sample size or small sample size. After the coefficient is modified, the estimation accuracy is improved obviously, which indicates that the accuracy of the coefficient is very important.

Figure-2. Comparison results of simulated data set and real data set



4. Conclusion

The incremental method proposed by Hall, *et al.* [1] can effectively solve the problem of large data samples or constantly updated data. However, the coefficient of the sample covariance matrix calculation formula has errors. In this paper, the coefficients with errors are modified. The experimental results show that there is a big difference between the coefficients a and b. The simulation study and real data analysis show that the modified update form has higher estimation accuracy, which indicates that the accuracy of the coefficients has a significant impact on the estimation error.

Appendix

The R code of the modified increment method is presented. (modified increment-code.docx)

```
## Simulation
rm(list=ls(all=TRUE))
library(MASS)
library(matrixcalc)
N=16;p=8
mu0=as.matrix(runif(p,0))
sigma0=as.matrix(runif(p,0,10))
ro=as.matrix(c(runif(round(p/2),-1,-0.5),runif(p-round(p/2),0.5,1)))
R0=ro%*%t(ro);diag(R0)=1
Sigma0=sigma0%*%t(sigma0)*R0
data=mvrnorm(N,mu0,Sigma0)

x<-as.matrix(data)
N0<-round(0.5*N)
p<-ncol(x)
xbar<-colMeans(x[1:N0,])
CZ=cov(scale(x))
C<-cov(x[1:N0,])
lambda<-eigen(C)$values
U<-eigen(C)$vectors
C1=C2=C

for (i in (N0+1):N){
xbar<-((i-1)/i)*xbar+(1/i)*x[i,]
xcenter<-t(t(x[i,]-xbar))
C1<-(1/i)*C1+((i-1)/(i^2))*xcenter%*%t(xcenter)
C2<-(1/i)*C2+((i-1)^2/(i^3))*xcenter%*%t(xcenter)
}
MSE1=frobenius.norm(C1-CZ);MSE1
MSE2=frobenius.norm(C2-CZ);MSE2

## Real data set
data<-read.csv("C:/data/Immunotherapy.csv")
x<-as.matrix(data)
N<-nrow(x)
N0<-round(N/2)
p<-ncol(x)
en<-matrix(rep(1),1,p)
xbar<-colMeans(x[1:N0,])
CZ=cov(scale(x))
C<-cov(x[1:N0,])
lambda<-eigen(C)$values
U<-eigen(C)$vectors
C1=C2=C

for (i in (N0+1):N){
xbar<-((i-1)/i)*xbar+(1/i)*x[i,]
xcenter<-t(t(x[i,]-xbar))
C1<-(1/i)*C1+((i-1)/(i^2))*xcenter%*%t(xcenter)
C2<-(1/i)*C2+((i-1)^2/(i^3))*xcenter%*%t(xcenter)
}
MSE1=frobenius.norm(CZ-C1)^2/(p^2);MSE1
MSE2=frobenius.norm(CZ-C2)^2/(p^2);MSE2
```

References

[1] Hall, P. M., Marshall, A. D., and Martin, R. R., 1998. "Incremental eigenanalysis for classification." In *British Machine Vision Conference*. pp. 286–295.

[2] Sharma, L. N., Tripathy, R. K., and Dandapat, S., 2015. "Multiscale energy and eigenspace approach to detection and localization of myocardial infarction." In *IEEE Transactions on Biomedical Engineering*. pp. 1827-1837.

[3] Li, J., Chen, X., Yi, W., Wei, L., and Yu, D., 2016. "Eigenspace-based generalized sidelobe canceler beamforming applied to medical ultrasound imaging." *Sensors*, vol. 16, p. 1192.

[4] Ye, J. H., Guo, Q., Y., Jiang, A., W., and X., L., 2021. "Cross-age face recognition method based on direct sum of Eigenspace. (Engineering Science), 1-6 in Chinese." *Journal of Zhengzhou University (Engineering Science)*, pp. 1-6.

[5] Wen, Z., Chao, Y., Xin, L., and Yin, Z., 2016. "Trace-penalty minimization for large-scale eigenspace computation." *Journal of Scientific Computing*, vol. 66, pp. 1175-1203.

[6] Ren, Z., Liang, Y., Wang, M., Yang, Y., and Chen, A., 2020. "An eigenspace divide-and-conquer approach for large-scale optimization." *Applied Soft Computing*, vol. 99, p. 106911.