

A Short Note on Resolving Singularity Problems in Covariance Matrices

Ezgi Ayyıldız¹, Vilda Purutçuoğlu Gazi¹ & Ernst Wit²

¹ Department of Statistics, Middle East Technical University, Ankara, Turkey

² Johann Bernoulli Institute, Groningen University, Groningen, The Netherlands

Correspondence: Vilda Purutçuoğlu Gazi, Department of Statistics, Middle East Technical University, Ankara 06800, Turkey. Tel: 90-312-210-5319. E-mail: vpurutcu@metu.edu.tr

Received: July 18, 2012 Accepted: August 8, 2012 Online Published: September 18, 2012

doi:10.5539/ijsp.v1n2p113 URL: <http://dx.doi.org/10.5539/ijsp.v1n2p113>

Abstract

In problems where a distribution is concentrated in a lower-dimensional subspace, the covariance matrix faces a singularity problem. In downstream statistical analyzes this can cause a problem as the inverse of the covariance matrix is often required in the likelihood. There are several methods to overcome this challenge. The most well-known ones are the eigenvalue, singular value, and Cholesky decompositions. In this short note, we develop a new method to deal with the singularity problem while preserving the covariance structure of the original matrix. We compare our alternative with other methods. In a simulation study, we generate various covariance matrices that have different dimensions and dependency structures, and compare the CPU times of each approach.

Keywords: singularity problems in covariances, matrix decomposition, diffusion matrix

1. Introduction

The high dimensional variances' structures are commonly observed in many mathematical sciences from linear algebra and statistical calculations to financial and genomic data analysis. For instance in multivariate statistical analysis, the major concern is either the hypothesis testing or the inference of population parameters that are based on mainly high dimensional matrices where the covariances are typically used in the associated expressions (Kutner et al., 2005). Alternatively in microarray studies within the genomic data analysis (Emmert-Streib & Dehmer, 2008) we need to deal with the intensities from hundreds of genes simultaneously in which the mean and variance estimates of genes are described by high dimensional matrices. When working on such underlying high dimensional covariance structures, we may face with the singularity problem if the distribution of the dataset is concentrated on a lower-dimensional subspace. In other words there are linear relationships between columns or rows of the covariance matrix, resulting in zero determinant in its calculation. This also means that the inverse of the underlying matrix becomes intractable, which leads to a computational problem in the likelihood. For instance when we work on a singular covariance matrix and the distribution function of the data is expressed via this matrix such as the density of the multivariate normal distribution, the associated likelihood function becomes intractable.

There are a number of methods to unravel the underlying challenge. The most well-known ones can be listed as the eigenvalue decomposition, singular value decomposition, and the Cholesky decomposition. In this study, we suggest an alternative method for these major approaches to solve the singularity problem. We perform our suggested method in different covariance matrices with distinct dimensions and singularity proportions and we compare our results with the other well-known methods in terms of computational time. For the evaluation, we use Monte Carlo methods. In Section 2 we describe the problem. In Section 3 we initially explain each method explicitly and represent our motivation for proposing a new approach against the singularity. Then in Section 4 we report our comparative analyzes and interpret the results. Finally in Section 5, we summarize the outcomes and discuss the future works.

2. Motivation

In finance, genomics and several other fields, the aim is to explain the change in state of a stock, a gene or other variable by applying a deterministic drift term plus a stochastic diffusion term, involving a Brownian motion. In inferring multivariate stochastic differential equations (Golightly & Wilkinson, 2005; Purutçuoğlu & Wit, 2008)

of the form

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dB_t, \quad (1)$$

the aim is to obtain estimates of the drift μ and diffusion matrix σ . The inference problem of μ is relatively straightforward, whereas the inference for σ can involve computational problems. In fact, especially in high-dimensional problems, the term $\sigma(X_t, t)$ can be computationally singular, or even exactly singular for particular values of X_t and t . In practice, this can occur when the number of financial instruments increases or when the number of genes included in the model becomes large. Also as the system converges to its stationary distribution, we are often faced with the problem of linear dependency between the state space quantities that causes a singularity in diffusion matrix. For instance in stochastic volatility model (Chib et al., 2006) as well as financial forecasting analysis (Andersen et al., 1999), also high dimensional inference problems in biochemical networks (Puruçuoğlu & Wit, 2008) or stochastic simulation of large biological systems (Puruçuoğlu & Wit, 2006) such types of problems are typically observed.

Although there are a number of alternative approaches to deal with the singularity problem, none of them explicitly deals with the actual structure of the linear relationship between the states. Particularly, in Bayesian inference, we may have partial observations of some states and the information about all the states should be used for inference about diffusion. In that case, simply focussing on a non-singular subspace of the states, may result in information loss. Instead, we propose a method that considers a lower dimensional, non-singular diffusion submatrix and explicitly models the linear relation between its dependent and independent terms.

3. Method

The singularity problem in covariance-like matrices is one of the common challenges in the algebraic calculations. A number of methods exists to unravel this problem. In the first parts of this section, we briefly review current well-known approaches. Then we state an ongoing problem in the calculation of singularity and finally describe our proposal.

3.1 Eigenvalue Decomposition

Considering that A is a symmetric matrix, U denotes a matrix that includes the set of eigenvectors of A , and the diagonal matrix Λ stores the eigenvalues of A in the diagonal elements, we can write the following equality for every A , U , and Λ .

$$AU = U\Lambda. \quad (2)$$

Hereby the eigen-value decomposition of A can be found by

$$A = U\Lambda U^{-1}. \quad (3)$$

Accordingly the square root of A can be calculated by taking the square root of Λ via

$$A^{1/2} = U\Lambda^{1/2}U^{-1}. \quad (4)$$

If A is non-negative semi-definite, then the eigenvalue decomposition of this matrix always exists and the associated eigenvalues are always positive or zero. The correlation, covariance, and cross-product matrices can be given as examples of such types of matrices (Healy, 1986). In this decomposition, the structure of the matrix can disappear if the original matrix is singular and we need to convert this matrix into nonsingular form by reducing the dimension (Johnson & Wichern, 2002; Rencher, 2002).

3.2 Singular Value Decomposition

Let A be an $(m \times n)$ rectangular matrix which can be written as a product of three matrices, namely, an $(m \times m)$ orthogonal matrix U , an $(m \times n)$ diagonal matrix Λ , and the transpose of an $(n \times n)$ orthogonal matrix V . Here Λ is a matrix containing singular values in diagonal elements with descending order.

Accordingly the singular value decomposition (SVD) of A can be declared in terms of U , Λ , and V by

$$A = U\Lambda V^T \quad (5)$$

where the columns of U and the columns of V are orthogonal eigenvectors of AA^T and $A^T A$, respectively. Thus the square root of A can be written via

$$A^{1/2} = U\Lambda^{1/2}V^T. \quad (6)$$

Similar to the eigenvalue decomposition, the structure of the original matrix A can change if A is reconstructed in order to convert the matrix from singular to nonsingular form (Johnson & Wichern, 2002; Rencher, 2002).

3.3 Cholesky Decomposition

If A denotes a symmetric and positive definite matrix, the Cholesky decomposition of A can be found by an upper triangular matrix U having strictly positive diagonal entries such that

$$A = U^T U. \quad (7)$$

Here the matrix U is called the square root of A . If A is also symmetric, then $U^T = U$, hereby, $A = UU$. But if A is positive semi-definite, i.e. some eigenvalues are zero, we use a numerical tolerance in the decomposition of A . In this approach, similar to its previous alternatives, it cannot preserve the original structure of the matrix when the singularity problem is solved by this decomposition and a new nonsingular matrix is defined under the original dimension of A (Johnson & Wichern, 2002; Rencher, 2002).

3.4 New Method

We suggest a new updating regime to solve the singularity problem of a covariance matrix A , which enables us to reconstitute the covariance structure by working with a lower dimensional matrix A^* and its linear recovery matrix B . Let A be a $(n \times n)$ symmetric matrix of rank $p < n$. Without loss of generality, assume that the first p columns are linearly independent. In practice, this is not a restriction, as the underlying states X (cf. Section 2) can always be reordered to achieve this. This means that the matrix A can be written as

$$A = [A_1 \ A_2],$$

where A_1 is an $(n \times p)$ and A_2 an $(n \times (n - p))$ matrix, for which the latter can be written as

$$A_2 = A_1 B,$$

for a $(p \times (n - p))$ matrix B , since A_2 is linearly dependent on A_1 by the assumption of rank p of A . In other words,

$$A = A_1 [I_p \ B]$$

for p dimensional identity matrix I_p . By decomposing A_1 in an upper $(p \times p)$ dimensional and a lower $((n - p) \times p)$ dimensional matrix, $A_1 = [A_{11} \ A_{12}]^T$, we can write A as

$$A = \begin{bmatrix} A_{11} & A_{11} B \\ A_{12} & A_{12} B \end{bmatrix}.$$

Given the symmetry of A , we get that $A_{12}^T = A_{11} B$ and so

$$A = \begin{bmatrix} A_{11} & A_{11} B \\ B^T A_{11}^T & B^T A_{11}^T B \end{bmatrix}. \quad (8)$$

The crucial element of the decomposition in (8) is that the distribution can be described by means of the $(p \times p)$ non-singular matrix A_{11} , whereas the matrix B of the linear coefficients allows us to reconstitute the linearly dependent elements. The steps of the algorithm for matrix A can be also listed as below.

- 1) Each linearly dependent column $A = [A_1 \ A_2]$ is identified by checking the columns of A from left to right.
- 2) The dependent columns, A_2 , totally p , are described as the linear combination of independent columns.
- 3) A new $(n - p) \times (n - p)$ dimensional covariance matrix A^* is defined by eliminating p dependent columns and rows from $A_{n \times n}$.

For example, if this method is used in MCMC (Monte Carlo Markov Chain) inference of stochastic differential equations, a singular diffusion matrix can be first transformed into a non-singular and lower dimensional form and the diffusion matrix can be updated under such lower dimensional space. Then the updated and independent columns are re-applied in the generation of the omitted columns, which are eliminated previously due to the dependency, by using the linear relations encoded in matrix B .

3.4.1 Numerical Example

As an implementation of the new method we give the following numerical example by applying Equation (1). Let the total number of substrates in the system be 3, $n = 3$, and the time step dt is taken as $dt = 0.1$ unit. Also the drift term μ and diffusion matrix σ set to $\mu = [3 \ 5 \ 7]$ and

$$\sigma = \begin{bmatrix} 4 & 16 & 0 \\ 16 & 64 & 0 \\ 0 & 0 & 9 \end{bmatrix},$$

respectively. By controlling the column of σ from left to right, we see that the second column of σ is four times of its first column, thereby the corresponding B vector which shows the linear dependence between the second column and the remaining columns is $B = [4 \ 0]$. In order to get a non-singular submatrix, we decompose the diffusion matrix σ as

$$\sigma = [\sigma_1 \ \sigma_2],$$

where

$$\sigma_1 = \begin{bmatrix} 4 & 0 \\ 0 & 9 \\ 16 & 0 \end{bmatrix} \text{ and } \sigma_2 = \begin{bmatrix} 16 \\ 0 \\ 64 \end{bmatrix}.$$

Since σ_2 is linearly dependent on σ_1 , the following equality is satisfied.

$$\sigma_2 = \sigma_1 B = \begin{bmatrix} 4 & 0 \\ 0 & 9 \\ 16 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix}.$$

Hereby, we can obtain an upper (2×2) dimensional matrix and a 2-dimensional vector by decomposing $\sigma_1 = [\sigma_{11} \ \sigma_{12}]^T$ via

$$\sigma_{11} = \begin{bmatrix} 4 & 0 \\ 0 & 9 \end{bmatrix} \text{ and } \sigma_{12} = [16 \ 0].$$

Hence we can generate the non-singular and lower dimensional matrix σ^* by $\sigma^* = \sigma_{11}$. So in order to obtain $\Delta_j = \sigma^{*1/2} \times dB_t$ according to Equation (1), resulting in the calculation of the diffusion part for the independent terms changed by Brownian motion, we generate two random values from normal distribution with mean zero and variance as the time step 0.1, i.e. $dB_t \sim N(0, 0.1)$. Then we multiply them by $\sigma^{*1/2}$. The values of dB_t are found as $dB_t = [-0.524 \ -0.251]$ from the R programme language with *rnorm* function. Accordingly Δ_j associated to the first and third substrates are obtained by $\Delta_{j=1,3} = \sigma^{*1/2} \times dB_t = (-1.048, -0.753)$. On the other hand for the update of the dependent term, i.e. the corresponding value of the second term, we calculate $\Delta_{i=2} = B \times \Delta_j = [4 \ 0] \times \begin{pmatrix} -1.048 \\ -0.753 \end{pmatrix} = -4.192$. And finally the change in state based on the diffusion model is computed as follows.

$$\begin{aligned} \Delta X_t &= \mu \times dt + \Delta \\ &= [3 \ 5 \ 7] \times 0.1 + [-1.048 \ -4.192 \ -0.753] \\ &= [-0.748 \ -3.692 \ -0.053]. \end{aligned}$$

In the further analysis, for instance, if the result is used for a simulation purpose, this change is applied for the update of the system in the next state. Hereby the new state X_{new} can be generated by adding the current state X_{old} to the change ΔX_t , i.e. $X_{new} = X_{old} + \Delta X_t$.

4. Simulation Study

In order to compare the performance of each method, we use simulated datasets and generate different covariance matrices which are symmetric and singular. In the matrix generation, firstly, we produce independent columns one by one from left to right. Then these columns are used to construct the linearly dependent columns. From the empirical studies we observe that the location of dependent and independent columns does not make any difference in our method since whatever the order of the dependent and independent columns, we always check each column of matrices from left to right sequentially and compare whether each proposal column is linearly dependent on previous columns. Hereby to obtain the symmetric matrix from the underlying singular matrix, the covariance matrix is calculated. These matrices are generated under two different scenarios. In the first plan, we generate

matrices whose 80 % of the columns are independent and the remaining 20 % of columns has linear dependency. Then in the second scenario, we use matrices whose 80 % of columns are linearly dependent, and the remaining ones are taken as independent. Under both scenarios, we apply matrices owing different dimensions, namely, 50 by 50, 100 by 100, 400 by 400, 500 by 500, and 750 by 750. In all computational works, we carry in the R programming language version 2.10.0 and execute our codes on Core 2 Duo 2.0 GHz processor.

To compare the performance of alternative methods with our proposal approach, we check the CPU (Central Processing Unit) times. In order to estimate the CPU of each method, we iterate the algorithm 1000 times for each matrix structure with distinct dimensions and take the mean of the underlying Monte Carlo outputs. However because of the time restriction and computational inefficiency, we iterate the algorithm 250 times for the matrix dimension 750 by 750. In these calculations, we record the time only to decompose singular matrices for eigenvalue, singular value, and Cholesky methods, and keep the time to both decompose the singularity and detect the linear relationship between columns for our suggested approach.

Table 1 shows CPU's of matrices with 4 different sizes under the first scenario. From the results it is seen that although the new method has more computational steps, there is no difference in CPU's for low like (50×50) and moderately high dimensional-matrices such as the matrices under (100×100) and (400×400) dimensions. Whereas if we consider very high dimensions like (500×500) or (750×750), the new method slightly loses more time with respect to other methods under both scenarios. For instance the CPU's for the (750×750) matrix under the first scenario with 250 iterations are 0.008, 0.008, 0.002, and 2.147 for the eigenvalue, singular value, Cholesky, and our new method, respectively. Similarly the CPU's for this matrix-dimension under the second scenario are found as 0.014, 0.014, 0.003, and 0.785 for the same order of decomposition's methods. On the other hand when we increase the number of dependent columns under the same dimension as seen in Table 2, we observe that all methods use less CPU. However the CPU of our method decreases more sharply and its becomes the most efficient method in computation when both dependency and dimension are big. This findings can be interpreted that our new method can be mainly preferable when the singularity becomes a severe challenge for the matrix.

Table 1. Comparison of total CPU's under 1000 Monte Carlo runs for eigenvalue, singular value, and Cholesky decompositions with new method based on different dimensional matrices under the first scenario

Methods	Dimensions			
	(50 × 50)	(100 × 100)	(400 × 400)	(500 × 500)
Eigenvalue decomposition	0.05	0.03	34.86	27.28
Singular value decomposition	0.03	0.02	13.11	30.72
Cholesky decomposition	0.01	0.03	15.81	27.11
Our method	0.01	0.02	37.68	265.74

Table 2. Comparison of total CPU's under 1000 Monte Carlo runs for eigenvalue, singular value, and Cholesky decompositions with new method based on different dimensional matrices under the second scenario

Methods	Dimensions			
	(50 × 50)	(100 × 100)	(400 × 400)	(500 × 500)
Eigenvalue decomposition	0.03	0.03	2.85	34.6
Singular value decomposition	0.02	0.05	1.47	29.35
Cholesky decomposition	0.03	0.04	0.09	23.48
Our method	0.05	0.05	0.94	9.44

5. Conclusion and Discussion

In this study, we have developed a new method to deal with the singularity problem of covariance matrix. We have explained the most well-known methods that are performed to overcome this challenge such as the eigenvalue, singular value, and Cholesky decompositions. Moreover we have generated different covariance matrices with distinct dimensions and dependency structures. Then we have used these matrices to check the performance of both our new method and the listed benchmark approaches in terms of their computational times. These computational times cover the decomposition of matrix with all suggested methods and also the time to find the linear relations of the columns for only our new method.

From the results we have observed that all methods have the same CPU's for low and moderately high dimensional

matrices. But when the dimension increases and the singularity is not seen in the majority of the columns, our approach is comparable to the other methods. However if we also raise the percentage of dependent columns, i.e. the singularity becomes severe in realistically large matrices, our proposal method becomes more efficient. Moreover, since it enables us to keep the same structure of the original entries of matrix by saving the linear relationship between columns, it can be more effectively applied in the reconstruction of the nonsingular version of the original matrix as needed in simulation studies (Purutçuoğlu & Wit, 2006) or modelling of the complex network structure by stochastic differential equation (Golightly & Wilkinson, 2005) or implementing the diffusion model in financial stochastic volatility models (Chib et al., 2006). Because in this approach, the preserved information of the linear relationship in the actual matrix can be used to regenerate eliminated columns. We believe that apart from these exemplified scenarios of singularities, our proposal decomposition algorithm can be also helpful for other singularity problems which we may face with in linear algebra and multiple statistical analyses.

References

- Andersen, T. G., Bollerslev, T., & Lange, S. (1999). Forecasting financial market volatility: Sample frequency vis-à-vis forecast horizon. *Journal of Empirical Econometrics*, 6(5), 457-477. [http://dx.doi.org/10.1016/S0927-5398\(99\)00013-4](http://dx.doi.org/10.1016/S0927-5398(99)00013-4)
- Chib, S., Nardari, F., & Shephard, N. (2006). Analysis of high dimensional multivariate stochastic volatility models. *Journal of Econometrics*, 134, 341-371. <http://dx.doi.org/10.1016/j.jeconom.2005.06.026>
- Emmert-Streib, F., & Dehmer, M. (2008). *Analysis of microarray data: A network-based approach*. Weinheim, Wiley-VCH Verlag.
- Golightly, A., & Wilkinson, D. J. (2005). Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61(3), 781-788. <http://dx.doi.org/10.1111/j.1541-0420.2005.00345.x>
- Healy, M. J. R. (1986). *Matrices for statistics*. Oxford: Oxford University Press.
- Johnson, R. A., & Wichern, D. W. (2002). *Applied multivariate statistical analysis*. Upper Saddle River, N. J., Prentice Hall.
- Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). *Applied linear statistical models*. Upper Saddle River, N. J., Prentice Hall.
- Purutçuoğlu, V., & Wit, E. (2006). Exact and approximate stochastic simulation of the MAPK pathway and comparisons of simulations' results. *Journal of Integrative Bioinformatics*, 3, 231-243. <http://dx.doi.org/10.2390/biecoll-jib-2006-38>
- Purutçuoğlu, V., & Wit, E. (2008). Bayesian inference for the MAPK/ERK pathway by considering the dependency of the kinetic parameters. *Bayesian Analysis*, 3(4), 851-886. <http://dx.doi.org/10.1214/08-BA332>
- Rencher, A. C. (2002). *Methods of multivariate analysis*. New York: John Wiley and Sons.