



Munich Personal RePEc Archive

The Monte Carlo method to find eigenvalues and eigenvectors

Ciuiu, Daniel and Costinescu, Cristian

Technical University of Civil Engineering, Bucharest, Romania,
Romanian Institute for Economic Forecasting, Technical University
of Civil Engineering, Bucharest, Romania

January 2008

Online at <https://mpra.ub.uni-muenchen.de/15362/>
MPRA Paper No. 15362, posted 23 May 2009 17:55 UTC

THE MONTE CARLO METHOD TO FIND EIGENVALUES AND EIGENVECTORS

Daniel Ciuiu

*Technical University of Civil Engineering, Bucharest, Romania
E-mail: dciuiu@yahoo.com*

Cristian Costinescu

*Technical University of Civil Engineering, Bucharest, Romania
E-mail: ccostinescu@yahoo.com*

Abstract: In this paper we apply the Monte Carlo method to find the eigenvalues and the eigenvectors of a k -symmetric matrix A . At first we add to the main diagonal of A a real number large enough to obtain a covariance matrix B and we take into account that the minimum sum of the squares in the principal components regression (PCR) is given by the corresponding eigenvector of the minimum eigenvalue of B .

Mathematics Subject Classification (2000): 62H25, 65C05, 34L15

Key words: Principal components regression, the Monte Carlo method, eigenvalues, eigenvectors

1. Introduction

Let be n points in \mathbf{R}^p : $X^{(1)}, \dots, X^{(n)}$. The orthogonal linear variety of the dimension k ($0 < k < p$) is that linear variety with the minimum sum of the squares of Euclidean distances. We know (see [9]) that this linear variety is generated by the eigenvectors of the sample covariance matrix corresponding to the first maximum k eigenvalues, and contains the gravity center of the given n points. These eigenvectors are called principal components, and for that the orthogonal regression is called also principal components regression (PCR). The principal components analysis is used in [10] to simplify the computations in the discriminant analysis by using the Kolmogoroff distance. There are many algebraic problems solved by using the Monte Carlo method instead of numerical methods, many of them using stochastic models. Stochastic models for solving some problems were used among others by Ermakov (see [5]) and Văduva (see [11]).

Let $A=(A_{ij})$ be a k -matrix and $x=(x_i), f=(f_i)$ be k -vectors. For solving the linear system

$$x = A \cdot x + f \quad (1)$$

Ermakov uses an ergodic Markov chain with k states, where k is the dimension of the system. The transition probabilities of this Markov chain are 0 for the null elements of A and non-zero values in the contrary case. We consider also an arbitrary vector h and an initial distribution $p = (p_i)_{i=1,k}$ with non-zero values at the same positions. Using a trajectory of this ergodic Markov chain Ermakov estimates the scalar product $\langle h, \bar{x} \rangle$, where \bar{x} is the solution of the system (1).

Văduva (see [11]) uses, oposite Ermakov, an absorbing Markov chain with $k+1$ states instead of an ergodic Markov chain with k states. The values P_{ij} for $1 \leq i, j \leq k$ are built in the same way as in Ermakov, but the sums of the transition probabilities from a state $i = \overline{1,k}$ to one of these states become less than 1. The differences to 1 are the probabilities to move to the state $k+1$ (absorption). Using N independent trajectories with the initial transient state i_0 Văduva estimates x_{i_0} .

In fact between the ergodic Markov chains and the absorbing Markov chains there exists a connection (see [7]).

In [2] we use a Jackson queueing network (see [6]) to solve some linear systems of equations. A Jackson queueing network is an open network with k nodes such that the interarrival time in the node i from outside the network is $\exp(\lambda_i)$, the service time at the node i is $\exp(\mu_i)$, and after he finishes his service at the node i , a customer goes to the node j with the probability P_{ij} or lives the network with the probability P_{i0} . The arrivals from outside network are set according to the right sides, and the transition probabilities are set according to the system matrix A (see [2]). In [3] we have solved by the Monte Carlo method the nonlinear equation in σ

$$A^*(\mu(1-\sigma)) = \sigma, \quad (2)$$

where A^* is the moments generating function of the interarrival times density function a (i.e. its Laplace transformation):

$$A^*(\zeta) = \int_0^{\infty} e^{-\zeta t} \cdot a(t) dt. \quad (2')$$

We generate (see [3]) a $G/M/1$ queueing system (see [8]) with the interarrival times density function a and the service times $\exp(\mu)$. We divide the simulation period t into m periods when we have no arrival and no service finalization. We estimate the average number of customers in the system, and, using this estimation, we estimate a solution of the nonlinear equation (2) if we know an analytical formula for A^* . In the contrary case, we solve an integral equation in the same way (see [3]).

2. Finding the eigenvalues and the eigenvectors

Because the orthogonal regression hyperplane is perpendicular to the principal component we can obtain the eigenvalues and the eigenvectors of a k -symmetric matrix A as follows: first we add to the main diagonal a real number large enough to obtain a strong diagonal dominant matrix B . We can choose

$$x = \max_{i=1,k} \left(\sum_{j \neq i} |A_{ij}| - A_{ii} \right) + 1. \quad (3)$$

The obtained matrix B is positive defined with the same eigenvectors, but its eigenvalues are greater than those of A by x from (3). We generate 1000 vectors distributed $N(0, B)$ using the following algorithm (see [11]) that generates a normal vector $N(\mu, \Sigma)$, denoting by I_k the k -unit matrix.

Algorithm normal

Compute L left-diagonal by the Cholesky factorization $\Sigma = L \cdot L^T$.

Generate the vector $Z \sim N(0, I_k)$ (k numbers $N(0, 1)$).

$$X = \mu + L \cdot Z$$

return X

end.

We generate also 1000 vectors uniform on the k -sphere. For this we can use the following algorithm (see [11]) that generates a uniform vector on the domain $D \subset \mathbb{R}^k$ by the rejection method:

Algorithm uniform

Consider the cartesian product I of the intervals $[a_i, b_i]$ such that $D \subset I$.

repeat

Generate a k -vector v uniform on I .

until $v \in D$

return v

end.

We generate first the vectors in the interior of the k -disc, and we find the normalized vectors.

Another algorithm to generate the uniform vectors on the sphere is to use the spherical coordinates: we generate the first $k-2$ spherical coordinates uniform in $[-\frac{\pi}{2}, \frac{\pi}{2}]$, and the last one uniform in $[0, 2 \cdot \pi]$. Consider the normal vectors $X^{(i)} = (X_1^{(i)}, \dots, X_k^{(i)})^T$ and the uniform vectors $v^{(i)} = (v_1^{(i)}, \dots, v_k^{(i)})^T$ with $i = \overline{1, 1000}$. For each $v = v^{(i)}$ we compute the sum

$$\sum_{i=1}^{1000} \left(\sum_{j=1}^k v_j \cdot X_j^{(i)} \right)^2 \quad (4)$$

and the corresponding eigenvector of the minimum eigenvalue is the vector v for that the above sum is minimum. The corresponding eigenvalue is this minimum sum divided by 1000 (if we change the axes by the principal components and the origin by the gravity center, the sum becomes the minimum variance multiplied by the number of points). If we have found $j < k-1$ eigenvalues of B and the corresponding eigenvectors, we apply first a rotation with a matrix having the first j columns equal to these eigenvectors. The other columns are obtained by adding the vectors of the canonic basis $\{e_1, \dots, e_k\}$ and using the Gramm-Schmidt algorithm. If the computed eigenvalues are $\lambda_1, \dots, \lambda_j$ the new covariance matrix is

$$\tilde{B} = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_j & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & C_{1,1} & \dots & C_{1,k-j} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & C_{k-j,1} & \dots & C_{k-j,k-j} \end{pmatrix}, \quad (5)$$

where C is a $(k-j) \times (k-j)$ matrix with the eigenvalues equal to the other $k-j$ eigenvalues of B .

We apply the above algorithm to find an eigenvalue of C and the corresponding eigenvector if we consider for the normal vectors the last $k-j$ components in the new coordinates system (after the rotation). We do the same with the uniform vectors, but for null vectors we set the first component to 1, and the other vectors are normalized. After the computation of an eigenvector of C we set the first j components to 0 and we make the inverse rotation to obtain a new eigenvector of B .

The last eigenvalue of B is the difference between $Trace(B)$ and the sum of the other eigenvalues, and the last eigenvector is perpendicular to the others. The matrix A has the same eigenvectors and its eigenvalues are less than those of B by the added constant used.

Example 1. Consider the matrix $A = \begin{pmatrix} -1.25 & 0.33 & 2.5 \\ 0.33 & -0.75 & 1.67 \\ 2.5 & 1.67 & 3 \end{pmatrix}$. If we apply the Jacobi rotation method

we obtain the eigenvalues $\begin{pmatrix} -2.5242 \\ -1.13954 \\ 4.66374 \end{pmatrix}$ and the eigenvectors on rows $\begin{pmatrix} 0.83937 & 0.28154 & -0.46497 \\ -0.38262 & 0.91362 & -0.1375 \\ 0.38609 & 0.29332 & 0.87458 \end{pmatrix}$.

If we generate the normal random variables using the central limit theorem and the uniform random vectors using the spherical coordinates we obtain the eigenvalues

$\begin{pmatrix} -2.58369 \\ -1.01904 \\ 4.60273 \end{pmatrix}$ and the eigenvectors on rows $\begin{pmatrix} 0.80314 & 0.35492 & -0.47854 \\ -0.43628 & 0.89734 & -0.06669 \\ 0.40574 & 0.26234 & 0.87553 \end{pmatrix}$.

If we generate the normal random variables using the central limit theorem and the uniform random vectors using the rejection method we obtain the eigenvalues

$\begin{pmatrix} -2.36353 \\ -1.11621 \\ 4.47974 \end{pmatrix}$ and the eigenvectors on rows $\begin{pmatrix} -0.84454 & -0.33855 & 0.4149 \\ -0.4363 & 0.88425 & -0.16657 \\ 0.31048 & 0.32169 & 0.89449 \end{pmatrix}$.

If we generate the normal random variables using the Box-Muler method and the uniform random vectors using the spheric coordinates we obtain the eigenvalues

$$\begin{pmatrix} -2.59655 \\ -1.08296 \\ 4.67951 \end{pmatrix} \text{ and the eigenvectors on rows } \begin{pmatrix} -0.8001 & -0.31456 & 0.51078 \\ -0.42169 & 0.90053 & -0.10595 \\ 0.42665 & 0.30016 & 0.85316 \end{pmatrix}.$$

If we generate the normal random variables using the Box-Muler method and the uniform random vectors using the rejection method we obtain the eigenvalues

$$\begin{pmatrix} -2.4101 \\ -1.52726 \\ 4.93736 \end{pmatrix} \text{ and the eigenvectors on rows } \begin{pmatrix} -0.83582 & -0.28491 & 0.46929 \\ -0.38895 & 0.91058 & -0.1399 \\ 0.38746 & 0.29946 & 0.87189 \end{pmatrix}.$$

If we generate the normal random variables using the first Butcher method and the uniform random vectors using the spheric coordinates we obtain the eigenvalues

$$\begin{pmatrix} -2.43781 \\ -0.96465 \\ 4.40246 \end{pmatrix} \text{ and the eigenvectors on rows } \begin{pmatrix} -0.84056 & -0.25578 & 0.47753 \\ -0.37543 & 0.91054 & -0.17312 \\ 0.39053 & 0.32479 & 0.86139 \end{pmatrix}.$$

If we generate the normal random variables using the first Butcher method and the uniform random vectors using the rejection method we obtain the eigenvalues

$$\begin{pmatrix} -2.71196 \\ -1.58691 \\ 5.29887 \end{pmatrix} \text{ and the eigenvectors on rows } \begin{pmatrix} -0.74638 & -0.4036 & 0.52917 \\ 0.51564 & -0.85339 & 0.07641 \\ 0.42075 & 0.3299 & 0.84507 \end{pmatrix}.$$

3. Conclusions

Generally the Monte Carlo method estimates the solution of a problem using some moments of a generated random variable. In our case we use the variances of some normal variables: when we change the coordinates $\{e_1, \dots, e_k\}$ by the principal components, the random normal vectors $N(0, B)$ become normal with their components independent with the average 0 . In fact the estimators of these variances are S^2 , which are biased. If we notice on running the program that the estimated eigenvalues are generally smaller then those obtained by the numerical methods we can change them by S'^2 (unbiased).

References

- [1] Bratley, P., Fox, L.B. and Schrage, L.E.: *A Guide to Simulation*, Springer Verlag, 1987.
- [2] Ciuiu, D.: Solving Linear Systems of Equations and Differential Equations with Partial Derivatives by the Monte Carlo Method using Service Systems, *Analele Universității București* **1** (2004), 93-104.
- [3] Ciuiu, D.: A Method to Solve Some Nonlinear Equations by the Monte Carlo Method Using Queueing Systems, *Analele Universității București*, **1** (2006), 21-30.
- [4] Ciuiu, D.: *Queueing Networks*, Ph. D. Thesis, Bucharest University, 2006 (Romanian).
- [5] Ermakov, S.M.: *The Monte Carlo Method and Connected Problems*, Technical Publishing House, Bucharest, 1976 (Romanian).
- [6] Garzia, M., Garzia, R., Kiemele, M. and Lockhart, C.: *Network modeling, simulation and analysis*, Marcel Decker, 1990.
- [7] Iosifescu, M.: *Finite Markov Chains and Applications*, Technical Publishing House, Bucharest, 1977 (Romanian).
- [8] Kleinrock, L.: *Queueing Systems*, John Wiley and Sons Inc., 1975.
- [9] Saporta, G.: *Probabilités, analyse des données et statistique*, Editions Technip, Paris, 1990.
- [10] Mahjoub, S. and Saporta, G.: Une méthode de discrimination non paramétrique, *Revue de Statistique Appliquée*, **Vol. XLII, No. 2** (1994), 99-113.
- [11] Văduva, I.: *Simulation Models*, Bucharest University Publishing House, 2004 (Romanian).