faster than the original data source and contains the most recently accessed items. Caches can exist at all levels in architecture, but are often found at the level nearest to the front end, where they are implemented to return data quickly without taxing downstream levels.

At a basic level, a proxy server is an intermediate piece of hardware/software that receives requests from clients and relays them to the backend origin servers. Typically, proxies are used to filter requests, log requests. Proxies are also immensely helpful when coordinating requests from multiple servers, providing opportunities to optimize request traffic from a system-wide perspective. One way to use a proxy to speed up data access is to collapse the same (or similar) requests together into one request, and then return the single result to the requesting clients.

Using an index to access your data quickly is a well-known strategy for optimizing data access performance; probably the most well-known when it comes to databases. An index makes the trade-offs of increased storage overhead and slower writes for the benefit of faster reads.

Load balancers are a principal part of any architecture, as their role is to distribute load across a set of nodes responsible for servicing requests. Their main purpose is to handle a lot of simultaneous connections and route those connections to one of the request nodes, allowing the system to scale to service more requests by just adding nodes. In a distributed system, load balancers are often found at the very front of the system, such that all incoming requests are routed accordingly. Load balancers also provide the critical function of being able to test the health of a node, such that if a node is unresponsive or over-loaded, it can be removed from the pool handling requests, taking advantage of the redundancy of different nodes in your system [5].

The development of effective systems with fast access to large amounts of data is a very interesting topic, and there are many different approaches that give the right to form the architecture of systems in the early stages of development. When designing a distributed web-based systems there are always a number of difficulties, the solution of which will have to sacrifice some principles to make full use of others. Some useful ways to develop a scalable system are: the separation of functionality to services, the use of redundancy to address failures, the use of data partitioning. With the growth of applications often use to simplify the systems approach, the main ones are proxy, indexes, caches and load balancers.

REFERENCES

1. Abbott, Martin L. Scalability Rules: 50 Principles for Scalling Web Sites / Martin L. Abbott. – 1st Edition. – Addison-Wesley, 2011.
2. Abbott, Martin L. The Art of Scalability: Scalable Web Architecture, Processes, and Organizations or the Modern Enterprise / Martin L. Abbott. – 2st Edition. – Addison-Wesley, 2013.
3. Schlossnagle, T. Scalable Internet Architecture / T. Schlossnagle. – 1st Edition. – Developer's Library, 2010.
4. Mani Krishna, C. Fault-Tolerant Systems / C. Mani Krishna. – 1st Edition. – Elsevier, 2008.
5. The Architecture of Open Source Applications [Electronic resource]. – Mode of access: http://www.aosabook.org/en/. – Date of access: 29.09.2015;

UDC 510

SINGULAR DECOMPOSITION OF MATRIXES IN TASKS

OLGA AGAFONOVA, STEPAN EKHILEVSKIY, NINA GURYEVA
Polotsk State University,Belarus

For any real or complex (m×n) – matrix A with rank r of the matrix A^*A and AA^*, where A^* is obtained from the matrix A by transposition and replacement of components on the complex conjugate is, are symmetric or Hermit with rank r and dimension according to n and m. Note that they are non-negative. Therefore characteristic numbers of such matrixes are the real non-negative numbers.

We designate characteristic numbers of a matrix A^*A through $\rho_1^2, \rho_2^2, ..., \rho_n^2$, considering that $\rho_1^2 \geq \rho_2^2 \geq ... \geq \rho_n^2$ ($\rho_i \neq 0$ when *i=1, 2,…, r*).

It is known that the operator with a symmetric or Hermit matrix A^*A has orthonormal system of eigenvectors $e_1, e_2, \dots, e_n$ respectively on $\rho_1^2, \rho_2^2, ..., \rho_n^2$, then there is such vectors that $A^*Ae_i = \rho_i^2 e_i$, $i = 1, 2, \dots, n$, where

$$\left(e_i, e_j\right) = \begin{cases} 1, \text{ when } i = j, \\ 0, \text{ when } i \neq j \end{cases}.$$

This system of vectors is transferred by the operator with a matrix A to some orthogonal system of vectors $Ae_1, Ae_2, \dots, Ae_n$, and $|Ae_i| = \rho_i$. Therefore vector Ae_i different from null-vector if and only if when $\rho_i \neq 0$, that is when *i=1, 2,…, r*. Null-vector Ae_i is an eigenvector of the matrix AA^* on their own value ρ_i^2 . Therefore null-eigenvalues of matrixes A^*A and AA^* are the same with regard to their multiplicities and their number is equal to r. Multiplicity of null-eigenvalue of these matrices are , respectively, *n-r* and *m-r*. General count eigenvalues of the matrices A^*A and AA^* is $s' = \min(m, n)$.

Arithmetic values $\rho_1, \rho_2, ..., \rho_{s'}$ square roots of the total characteristic numbers of matrices A^*A and AA^* are called singular or main numbers of the matrix *A*.

Take for a basis in the space X_n orthonormal system of vectors $e_1, e_2, \dots, e_n$ eigenvectors of the matrix A^*A and construct orthonormal system of vectors

$$f_1 = \frac{Ae_1}{|Ae_1|} = \frac{Ae_1}{\rho_1}, \dots, f_r = \frac{Ae_r}{|Ae_r|} = \frac{Ae_r}{\rho_r}$$

We supplement the system of vectors any vectors $f_{r+1}, \dots, f_m$ to an orthonormal basis in the space Y_m . By construction, the vectors $f_1, \dots, f_m$ satisfy relations

$$Ae_i = \begin{cases} \rho_i f_i, \text{ when } i \le r, \\ 0, \text{ when } i > r. \end{cases} \tag{1}$$

Multiplying this equality on the left by A^*, considering that $A^*Ae_i = \rho_i^2 \cdot e_i$, we obtain the relations

$$A^* f_i = \begin{cases} \rho_i e_i, \text{ when } i \le r \\ 0, \text{ when } i > r \end{cases} \tag{2}$$

Orthonormal bases $e_1, e_2, \dots, e_n$ and $f_1, f_2, \dots, f_m$ of spaces X_n and Y_m , which are related to (1) and (2), are called singular bases. These vectors e_i are called right singular vectors of the matrix *A*, and the vectors f_k are called left singular vectors.

The operator, which has in the original basis of spaces X_n and Y_m (m×n)-matrix *A*, has next matrix in singular bases:

$$\Sigma = \begin{pmatrix} \rho_1 & 0 & \dots & 0 & 0 \\ 0 & \rho_2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \rho_r & 0 \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix} \tag{3}$$

By formula, which establishes a connection between the matrixes of the same operator in different bases, we have

$$A = Q \cdot \Sigma \cdot P^* \tag{4}$$

where $P = (e_1, e_2, \dots, e_n)$ – orthogonal or unitary matrix with order *n*, whose columns are the coordinate columns of the vectors $e_1, e_2, \dots, e_n$ in the original basis of space X_n , $Q = (f_1, f_2, \dots, f_m)$ – orthogonal or unitary matrix with order *m*, whose columns are the coordinate columns of the vectors $f_1, f_2, \dots, f_m$ in the original basis of the space Y_m .

Expansion (4) is called the singular decomposition of the matrix *A*. Note that any (m×n)-matrix has many different singular decompositions, which follow from a certain arbitrariness in the in the structure and systems of vectors $e_1, e_2, \dots, e_n$ and $f_1, f_2, \dots, f_m$.

Consider the following problem.

For a matrix *A* construct singular decomposition:

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Decision:

We first note that the matrix has dimension m×n = 3×4. We construct a symmetric matrix $A^* A$, where A^* is received by transposing the matrix *A*:

$$A^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix},\ A^* A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}$$

The characteristic polynomial of the resulting matrix is a polynomial of the form $\left|A^* A - \lambda E\right| = \lambda(\lambda - 4)(1 - \lambda)^2$

Its roots are $\lambda_1 = 4,\ \lambda_2 = \lambda_3 = 1,\ \lambda_4 = 0$. From here follows owing to equality $\lambda_i = \rho_i^2$, that $\rho_1 = 2,\ \rho_2 = 1,\ \rho_3 = 1,\ \rho_4 = 0$.

We construct a matrix Σ, considering that $\min(m, n) = \min(3, 4) = 3$:

$$\Sigma = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

To construct the matrix *P* with order *n* = 4 should build orthonormal system eigenvectors $e_1,\ e_2,\ e_3,\ e_4$ of matrix $A^* A$. For this we use the definition of eigenvector $A^* Ax = \lambda x$.

We have now the matrix

$$\left(A^* A - \lambda E\right) = \begin{pmatrix} 1-\lambda & 0 & 0 & 1 \\ 0 & 1-\lambda & 0 & 1 \\ 0 & 0 & 1-\lambda & 1 \\ 1 & 1 & 1 & 3-\lambda \end{pmatrix},$$

which allows to submit a matrix equation $\left(A^* A - \lambda E\right) X = 0$ in the form of equality:

$$\begin{pmatrix} 1-\lambda & 0 & 0 & 1 \\ 0 & 1-\lambda & 0 & 1 \\ 0 & 0 & 1-\lambda & 1 \\ 1 & 1 & 1 & 3-\lambda \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

We receive homogeneous system of the equations

$$\begin{cases} (1-\lambda)x_1 + x_4 = 0, \\ (1-\lambda)x_2 + x_4 = 0, \\ (1-\lambda)x_3 + x_4 = 0, \\ x_1 + x_2 + x_3 + (3-\lambda)x_4 = 0 \end{cases}$$

When $\lambda = 4$ the system has a fundamental system of solutions, which consists of one solution vector, for example, from vector $(1, 1, 1, 3)^{\mathrm{T}}$, normalizing which we have received $e_1 = \frac{1}{2\sqrt{3}}(1, 1, 1, 3)^{\mathrm{T}}$.

When $\lambda = 1$ the system has a fundamental system of solutions, which consists of two vectors, for example, from vectors $(1, -1, 0, 0)^{\mathrm{T}}$ and $(1, 0, -1, 0)^{\mathrm{T}}$, orthogonalizing and normalizing this system of vectors we have received $e_2 = \frac{1}{\sqrt{2}}(1, -1, 0, 0)^{\mathrm{T}},\ e_3 = \frac{1}{\sqrt{6}}(1, 1, -2, 0)^{\mathrm{T}}$.

When $\lambda = 0$ the system has a fundamental system of solutions, which consists of one solution vector, for example, from vector $(1,1,1,-1)^{\mathrm{T}}$, normalizing which we have received $e_4 = \frac{1}{2}(1,1,1,-1)^{\mathrm{T}}$.

From the columns coordinates of the vectors $e_1,\ e_2,\ e_3,\ e_4$ make a matrix

$$P = (e_1,\ e_2,\ e_3,\ e_4) = \begin{pmatrix} \frac{1}{2\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & \frac{1}{2} \\ \frac{1}{2\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & \frac{1}{2} \\ \frac{1}{2\sqrt{3}} & 0 & -\frac{1}{\sqrt{6}} & \frac{1}{2} \\ \frac{3}{2\sqrt{3}} & 0 & 0 & -\frac{1}{2} \end{pmatrix}.$$

Next construct vectors

$$f_1 = \frac{Ae_1}{|Ae_1|} = \frac{Ae_1}{\rho_1} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \cdot \frac{1}{2\sqrt{3}}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 3 \end{pmatrix} = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

$$f_2 = \frac{Ae_2}{|Ae_2|} = \frac{Ae_2}{\rho_2} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \cdot \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix},$$

$$f_3 = \frac{Ae_3}{|Ae_3|} = \frac{Ae_3}{\rho_3} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \cdot \frac{1}{\sqrt{6}}\begin{pmatrix} 1 \\ 1 \\ -2 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{6}}\begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}.$$

These vectors constitute a basis in the space Y_3.

From the columns coordinates of the vectors $f_1,\ f_2,\ f_3$ make a matrix

$$Q = (f_1,\ f_2,\ f_3) = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \end{pmatrix}.$$

and present a singular decomposition of the matrix A:

$$A = Q\Sigma P^* = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \end{pmatrix} \cdot \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{3}{2\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

Vectors $e_1,\ e_2,\ e_3,\ e_4$ and $f_1,\ f_2,\ f_3$ constitute singular bases in the spaces X_4 and Y_3. The correctness of the solution is checked by calculating the product of constructed matrices.

If we take the other basis vectors, such as $e_1 = \frac{1}{2\sqrt{3}}\left(1,1,1,3\right)^T$, $e_2 = \frac{1}{\sqrt{2}}\left(0,1,-1,0\right)^T$, $e_3 = \frac{1}{\sqrt{6}}\left(2,-1,-1,0\right)^T$, $e_4 = \frac{1}{2}\left(-1,-1,-1,1\right)^T$ and $f_1 = \frac{1}{\sqrt{3}}\left(1,1,1\right)^T$, $f_2 = \left(0,\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{6}}\right)^T$, $f_3 = \frac{1}{\sqrt{6}}\left(2,-1,-1\right)^T$, the singular decomposition of the matrix A is of the form:

$$A = Q\Sigma P^* = \begin{pmatrix} \frac{1}{\sqrt{3}} & 0 & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \end{pmatrix} \cdot \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{3}{2\sqrt{3}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

UDC 004.451.83

THE FRONT-END DESIGN OF CINEMA POSTER CONTROL SYSTEM OF THE REPUBLIC OF BELARUS

VICTOR DEMENTIEV, MARIA DEKANOVA
Polotsk State University, Belarus

The subject of development is cinema poster control system of the Republic of Belarus. The purpose of the work is creating a system for convenient management of cinema poster of the Republic of Belarus. In the article issues connected with a process of control system front-end design are considered.

The purpose of control system front-end development is to provide convenience for administrator's work with data base records and excluding possible errors in data base work. Thus, a particular thought should be given to graphical user interface development.

Generally most control systems contain several weak points. As a rule most CMS are developed to meet the common needs (articles adding, users moderating, etc.). That's why developers experience certain problems while different specific and unique modules adding to these CMS [4]. As a result the entire system becomes more bulky. Some opportunities of system aren't used because they are not needed for current demands of the user, but they require some certain amount of resources. Besides the style of common CMS is not up to date enough. Now Google tends to use the so called Material Design [3], which is similar to minimalism. This style is used in the development of control systems. This design language gets more and more popular nowadays and looks neat and accurate. There is no need in colourful pictures which may confuse the user, lose some important information or even not perceive it. The entire panel of administrator of the system should be implemented in a unified easy on the eye style. It should also be noticed that a lot of systems do not provide users with convenient work. For major business process tasks execution user has to carry out different not obvious steps in these systems. The current system should be absolutely clear and pellucid. It is neccessary to provide the ability of any cinema administrator to perform easily any action by clicking no more than two buttons. Special attention should be paid to ergonomics. Convenient location of elements allows to make the work easier and to decrease cognitive pressure.

As a development environment JetBrais PhpStorm 8 has been chosen. This integrated development environment has a perfect user interface and supports the latest versions of PHP. Besides, the PhpStorm gives perfect opportunities for comfortable development in different programming languages such as JavaScript, HTML и CSS [2].

The Skeleton library was taken as a foundation of CSS stylistics for building system [1]. This library gives minimal opportunities for starting a personal project development. That is why all major styles are created autographic.

Font Awesome library is used for display of icons [1]. It contains vector icons, which are integrated to font. Their advantage is that they ate able to change size without loss of quality. To say more different CSS styles may be applied to them, for example, changing of colour.

JQuery library is used for convenient work with JavaScript [2]. It gives an easy access to DOM elements of page. The most of POST/GET queries are processed using AJAX technology [2]. So, it decreases time of waiting for getting response, because only necessary data will load to a page. Other data will not be updated.