Universal algorithms for generalized discrete matrix Bellman equations with symmetric Toeplitz matrix

© S. N. Sergeev
Ecole Polytechnique, Paris, France

This paper presents two universal algorithms for generalized discrete matrix Bellman equations with symmetric Toeplitz matrix. The algorithms are semiring extensions of two well-known methods solving Toeplitz systems in the ordinary linear algebra.

Keywords: universal algorithms, matrix Bellman equations, Toeplitz matrix, semirings, idempotent semirings

§ 1. Introduction

As observed by B. A. Carré and R. C. Backhouse [2], [4], the Gaussian elimination without pivoting can be viewed as a prototype for some algorithms on graphs. M. Gondran [8] and G. Rote [21] made this observation precise by proving that the Gaussian elimination, under certain conditions, can be applied to the linear systems of equations over semirings. The notion of universal algorithm over semiring was introduced by G. L. Litvinov, V. P. Maslov and E. V. Maslova in [10], [15]. These papers are to be considered in the framework of publications [9], [11], [12], [17], [18], [20] of the Russian idempotent school, and more generally in the framework of idempotent and tropical mathematics, see [1], [5], [14], [16] and references therein. Essentially, an algorithm is called universal if it does not depend on the computer representation of data and on a specific realization of algebraic operations involved in the algorithm [15]. Linear algebraic universal algorithms include generalized bordering method, LU- and LDM-decompositions for solving matrix equations. These methods are basically due to B. A. Carré, see also [15].

It was observed in [13], [15] that universal algorithms can be implemented by means of objective-oriented programming supported by C++, MATLAB, Scilab, Maple and other computer systems and languages. Such universal programs can be instrumental in many areas including the problems of linear algebra, optimization theory, and interval analysis over positive semirings, see [13], [17], [18], [19].

1This work is partially supported by the RFBR-CRFN grant 11-01-93106
This paper presents new universal algorithms based on the methods of Durbin and Levinson, see [7], Sect. 4.7. These algorithms solve systems of linear equations with symmetric Toeplitz matrices. Our universal algorithms have the same complexity $O(n^2)$ as their prototypes which beats the complexity $O(n^3)$ of the LDM-decomposition method. All algorithms are described as MATLAB-programs (following the style of [7]), meaning that they can be actually implemented.

The author is grateful to G. L. Litvinov and A. N. Sobolevskii for drawing his attention to this problem and for valuable discussions.

§ 2. Semirings and universal algorithms

A set $S$ equipped with addition $\oplus$ and multiplication $\odot$ is a semiring (with unity) if the following axioms hold:

1) $(S, \oplus)$ is a commutative semigroup with neutral element $0$;

2) $(S, \odot)$ is a semigroup with neutral element $1 \neq 0$;

3) $a \odot (b \oplus c) = (a \odot b) \oplus (a \odot c)$, $(a \oplus b) \odot c = (a \odot c) \oplus (b \odot c)$ for all $a, b, c \in S$ (distributivity);

4) $0 \odot a = a \odot 0 = 0$ for all $a \in S$.

The notation $\odot$ will be sometimes omitted.

The semiring $S$ is called idempotent if $a \oplus a = a$ for any $a \in S$. In this case $\oplus$ induces the canonical partial order relation

$$a \preceq b \iff a \oplus b = b.$$  \hspace{1cm} (1)

The semiring $S$ is called complete (cf. [6]), if any subset $\{x_\alpha\} \subseteq S$ is summable and the infinite distributivity

$$c \odot (\bigoplus_\alpha x_\alpha) = \bigoplus_\alpha (c \odot x_\alpha),$$

$$\bigoplus_\alpha x_\alpha \odot c = \bigoplus_\alpha (x_\alpha \odot c),$$

holds for all $c \in S$ and $\{x_\alpha\} \subseteq S$. This property is natural in idempotent semirings and also in the theory of partially ordered spaces (cf. G. Birkhoff [3]) with partial order (1). Complete idempotent semirings are also called $a$-complete (cf. [11]).

Consider the closure operation

$$a^* = \bigoplus_{i=0}^\infty a^i.$$  \hspace{1cm} (2)
In the complete semirings it is defined for all elements. The property

\[ a^* = 1 \oplus aa^* = 1 \oplus a^*a, \]  \hspace{1cm} (3)

reveals that the closure operation is a natural extension of \((1 - a)^{-1}\).

We give some examples of semirings living on the set of reals \(\mathbb{R}\) totally ordered by \(\leq\): the semiring \(\mathbb{R}_+\) with customary operations \(\oplus = +\), \(\odot = \cdot\) and neutral elements \(0 = 0\) and \(1 = 1\); the semiring \(\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}\) with operations \(\oplus = \max\), \(\odot = +\) and neutral elements \(0 = -\infty\), \(1 = 0\); the semiring \(\mathbb{R}_{\max} = \mathbb{R}_{\max} \cup \{\infty\}\), which is a completion of \(\mathbb{R}_{\max}\) with the element \(\infty\) satisfying \(a \oplus \infty = \infty\) for all \(a\), \(a \odot \infty = \infty \odot a = \infty\) for \(a \neq 0\) and \(0 \odot \infty = \infty \odot 0 = 0\); the semiring \(\mathbb{R}_{\max,\min} = \mathbb{R} \cup \{\infty\} \cup \{-\infty\}\) with \(\oplus = \max\), \(\odot = \min\), \(0 = -\infty\), and \(1 = \infty\).

Consider operation (2) for the examples above. In \(\mathbb{R}_+\) the closure \(a^*\) equals \((1 - a)^{-1}\) if \(a < 1\) and is undefined otherwise; in \(\mathbb{R}_{\max}\) it equals 1 if \(a \leq 1\) and is undefined otherwise; in \(\mathbb{R}_{\max}\) we have \(a^* = 1\) for \(a \leq 1\) and \(a^* = \infty\) for \(a > 1\); in \(\mathbb{R}_{\max,\min}\) we have \(a^* = 1\) for all \(a\). Note that \(\mathbb{R}_{\max}\) and \(\mathbb{R}_{\max,\min}\) are a-complete, so the closure is defined for any element of these semirings.

The matrix operations \(\oplus\) and \(\odot\) are defined analogously to their counterparts in linear algebra. Denote by \(\text{Mat}_{mn}(S)\) the set of all \(m \times n\) matrices over the semiring \(S\). By \(I_n\) we denote the \(n \times n\) unity matrix, that is, matrix with 1 on the diagonal and 0 off the diagonal. As usual, we have \(AI_n = I_nA = A\) and \(A^0 = I_n\) for any \(A \in \text{Mat}_{nn}(S)\). The set \(\text{Mat}_{nn}(S)\) of all \(n \times n\) matrices is a semiring. Its unity is \(I_n\) and its zero is \(0_n\), the square matrix with all entries equal to 0. If \(S\) (and hence \(\text{Mat}_{nn}(S)\)) is complete, the closure \(A^*\) is defined for any matrix \(A\) and it satisfies (3). Note that if \(S\) is partially ordered, then \(\text{Mat}_{nn}(S)\) is ordered elementwise: \(A \leq B\) iff \(A_{ij} \leq B_{ij}\) for all \(i = 1, \ldots, m\) and \(j = 1, \ldots, n\). If \(S\) is idempotent and canonically ordered (1), then the elementwise order of \(\text{Mat}_{nn}(S)\) also satisfies (1).

The closure operation of matrices is important for the (discrete stationary) matrix Bellman equations

\[ X = AX \oplus B. \]  \hspace{1cm} (4)

If the closure of \(A\) exists and (3) holds, then \(X = A^*B\) is a solution of (4). In a-complete idempotent semirings the matrix \(A^*B\) is the least solution of equation (4) with respect to the canonical order (1).

Since \(A^*\) is a generalization of \((I - A)^{-1}\), the known universal algorithms for \(A^*\) are generalizations of the methods for matrix inverses, and the known algorithms for Bellman equations are generalizations of the methods for \(AX = B\). Further we consider the generalized bordering method (see also [2,4,8,15,21]).

Let \(A\) be a square matrix. Closures of its main submatrices \(A_k\) can be found inductively. The base of induction is \(A_1^*\), the closure of the first diagonal entry. Generally we represent \(A_{k+1}\) as

\[ A_{k+1} = \begin{pmatrix} A_k & g_k \\ h_k^T & a_{k+1} \end{pmatrix}, \]

assuming that we have found the closure of \(A_k\). In this representation, \(g_k\) and \(h_k\) are...
columns with $k$ entries and $a_{k+1}$ is a scalar. We also represent $A_{k+1}^*$ as
\[ A_{k+1}^* = \begin{pmatrix} u_k & v_k \\ w_k^T & u_{k+1} \end{pmatrix}. \]

Using (3) we obtain that
\[ u_{k+1} = (h_k^T A_k^* g_k \oplus a_{k+1})^*; \]
\[ v_k = A_k^* g_k u_{k+1}; \]
\[ w_k^T = u_{k+1} h_k^T A_k^*; \]
\[ U_k = A_k^* g_k u_{k+1} h_k^T A_k^* \oplus A_k^*. \] (5)

Consider the bordering method for finding the solution $x = A^* b$ to (4), where $X = x$ and $B = b$ are column vectors. Firstly, we have $x^{(1)} = A_1^* b_1$. Let $x^{(k)}$ be the vector found after $(k - 1)$ steps, and let us write
\[ x^{(k+1)} = \begin{pmatrix} z \\ x_{k+1} \end{pmatrix}. \]

Using (5) we obtain that
\[ x_{k+1} = u_{k+1} (h_k^T x^{(k)} \oplus b_{k+1}); \]
\[ z = x^{(k)} \oplus A_k^* g_k x_{k+1}. \] (6)

We have to compute $A_k^* g_k$. In the next section we show that this can be done very efficiently in the case when $A$ is symmetrical Toeplitz.

We also note that the bordering method described by (5) and (6) is valid more generally over Conway semirings, see [6] for definition.

§ 3. Universal algorithms for Toeplitz linear systems

Formally, a matrix $A \in \text{Mat}_{mn}(S)$ is called Toeplitz if there exist scalars $r_i$, $i = -n + 1, \ldots, 0, \ldots, n - 1$, such that $A_{ij} = r_{j-i}$ for all $i$ and $j$. Informally, Toeplitz matrices are such that their entries are constant along any line parallel to the main diagonal (and along the main diagonal itself). For example,
\[ A = \begin{pmatrix} r_0 & r_1 & r_2 & r_3 \\ r_{-1} & r_0 & r_2 & r_3 \\ r_{-2} & r_{-1} & r_0 & r_2 \\ r_{-3} & r_{-2} & r_{-1} & r_0 \end{pmatrix} \]
is Toeplitz. Such matrices are not necessarily symmetric. However, they are always persymmetric, that is, symmetric with respect to the inverse diagonal. This property
is algebraically expressed as $A = E_n A^T E_n$, where $E_n = [e_n, \ldots, e_1]$. By $e_i$ we denote the column whose $i$th entry is 1 and other entries are 0. The property $E_n^2 = I_n$ (where $I_n$ is the $n \times n$ identity matrix) implies that the product of two persymmetric matrices is persymmetric. Hence any degree of a persymmetric matrix is persymmetric, and so is the closure of a persymmetric matrix. Thus, if $A$ is persymmetric, then

$$E_n A^* = (A^*)^T E_n.$$  \hspace{1cm} (7)

Further we deal only with symmetric Toeplitz matrices. Consider the equation $y = T_n y \oplus r^{(n)}$, where $r^{(n)} = (r_1, \ldots, r_n)^T$, and $T_n$ is defined by the scalars $r_0, r_1, \ldots, r_{n-1}$ so that $T_{ij} = r_{|j-i|}$ for all $i$ and $j$. This is a generalization of the Yule-Walker problem [7]. Assume that we have obtained the least solution $y^{(k)}$ to the system $y = T_k y \oplus r^{(k)}$ for some $k$ such that $1 \leq k \leq n - 1$, where $T_k$ is the main $k \times k$ submatrix of $T_n$. We write $T_{k+1}$ as

$$T_{k+1} = \begin{pmatrix} T_k & E_k r^{(k)} \\ r^{(k)T} E_k & r_0 \end{pmatrix}.$$  \hspace{1cm} (8)

We also write $y^{(k+1)}$ and $r^{(k+1)}$ as

$$y^{(k+1)} = \begin{pmatrix} z \\ \alpha_k \end{pmatrix}, \quad r^{(k+1)} = \begin{pmatrix} r^{(k)} \\ r_{k+1} \end{pmatrix}.$$  \hspace{1cm} (9)

Using (6), (7) and the identity $T_k r^{(k)} = y^{(k)}$, we obtain that

$$\alpha_k = (r_0 \oplus r^{(k)T} y^{(k)})^* (r^{(k)T} E_k y^{(k)} \oplus r_{k+1}),$$
$$z = E_k y^{(k)} \alpha_k \oplus y^{(k)}.$$  \hspace{1cm} (10)

Denote $\beta_k = r_0 \oplus r^{(k)T} y^{(k)}$. The following argument shows that $\beta_k$ can be found recursively if $(\beta_{k-1}^*)^{-1}$ exists.

$$\beta_k = r_0 \oplus [r^{(k-1)T} r_k] \left( E_{k-1} y^{(k-1)} \alpha_{k-1} \oplus y^{(k-1)} \right) =$$
$$= r_0 \oplus r^{(k-1)T} y^{(k-1)} \oplus (r^{(k-1)T} E_{k-1} y^{(k-1)} \oplus r_k) \alpha_{k-1} =$$
$$= \beta_{k-1} \oplus (\beta_{k-1}^*)^{-1} \alpha_{k-1}^2.$$  \hspace{1cm} (11)

Existence of $(\beta_{k-1}^*)^{-1}$ is not universal, and this will make us write two versions of our algorithm, the first one involving (8), and the second one not involving it. We will write these two versions in one program and mark the expressions which refer only to the first version or to the second one by the MATLAB-style comments %1 and %2, respectively. Collecting the expressions for $\beta_k$, $\alpha_k$ and $z$ we obtain the following recursive expression for $y^{(k)}$:

$$\beta_k = r_0 \oplus r^{(k)T} y^{(k)}, \quad %2$$
$$\beta_k = \beta_{k-1} \oplus (\beta_{k-1}^*)^{-1} \alpha_{k-1}^2, \quad %1$$
$$\alpha_k = (\beta_k)^* \oplus (r^{(k)T} E_k y^{(k)} \oplus r_{k+1}),$$
$$y^{(k+1)} = \begin{pmatrix} E_k y^{(k)} \alpha_k \oplus y^{(k)} \\ \alpha_k \end{pmatrix}.$$  \hspace{1cm} (12)
Recursive expression (9) is a generalized version of the Durbin method for the Yule-Walker problem [7]. Using this expression we obtain the following algorithm.

**Algorithm 1.** The Yule-Walker problem for the Bellman equations with symmetric Toeplitz matrix.

```plaintext
function y = durbin(r_0, r)
    n = size(r) + 1
    y(1) = r_0 \odot r(1)
    \beta = r_0 \% 1 \\
    \alpha = r_0 \odot r(1)
    for k = 1 : n - 1
        \beta = r_0 \oplus r(1 : k) \odot y(1 : k) \% 2
        \beta = \beta \oplus (\beta^*)^{-1} \odot \alpha^2 \% 1
        \alpha = \beta^* \odot (r(k : -1 : 1) \odot y(1 : k) \oplus r(k + 1))
        z(1 : k) = y(1 : k) \oplus y(k : -1 : 1) \odot \alpha
        y(1 : k) = z(1 : k)
        y(k + 1) = \alpha
    end
end
```

Now we consider the problem of finding \( x^{(n)} = T_n^* b^{(n)} \) where \( T_n \) is as above and \( b^{(n)} = (b_1, \ldots, b_n) \) is arbitrary. We also introduce the column vectors \( y^{(k)} \) which solve the Yule-Walker problem: \( y^{(k)} = T_k^* r^{(k)} \). The main idea is to find the expression for \( x^{(k+1)} = T_{k+1}^* b^{(k+1)} \) involving \( x^{(k)} \) and \( y^{(k)} \). We write \( x^{(k+1)} \) and \( b^{(k+1)} \) as

\[
x^{(k+1)} = \left( \begin{array}{c} v \\ \mu_k \end{array} \right), \quad b^{(k+1)} = \left( \begin{array}{c} b_k^{(k)} \\ b_{k+1} \end{array} \right).
\]

Making use of the persymmetry of \( T_k^* \) and of the identities \( T_k^* b_k = x^{(k)} \) and \( T_k^* r_k = y^{(k)} \), we specialize expressions (6) and obtain that

\[
\mu_k = (r_0 \oplus r^{(k)T} y^{(k)})^* \odot (r^{(k)T} E_k x^{(k)} \oplus b_{k+1}),
\]

\[
v = E_k y^{(k)} \mu_k \oplus x^{(k)}.
\]

The coefficient \( r_0 \oplus r^{(k)T} y^{(k)} = \beta_k \) can be expressed again as \( \beta_k = \beta_{k-1} \oplus (\beta_{k-1}^*)^{-1} \odot (\alpha_{k-1})^2 \), if the closure \( (\beta_{k-1})^* \) is invertible. Using this we obtain the following recursive expression:

\[
\beta_k = r_0 \oplus r^{(k)T} y^{(k)}, \quad % 2
\]

\[
\beta_k = \beta_{k-1} \oplus (\beta_{k-1}^*)^{-1} \odot \alpha_{k-1}^2, \quad % 1
\]

\[
\mu_k = \beta_k^* \odot (r^{(k)T} E_k x^{(k)} \oplus b_{k+1}),
\]

\[
x^{(k+1)} = \left( \frac{E_k y^{(k)} \mu_k \oplus x^{(k)}}{\mu_k} \right).
\]

Expressions (9) and (10) yield the following generalized version of the Levinson algorithm for solving linear symmetric Toeplitz systems [7]:

**Algorithm 2.** Bellman system with symmetric Toeplitz matrix.
function \( y = \text{levinson}(r_0, r, b) \)
\( n = \text{size}(b) \)
\( y(1) = r_0^* \odot r(1); \quad x(1) = r_0^* \odot b(1); \)
\( \beta = r_0 \% 1 \)
\( \alpha = r_0^* \odot r(1) \)

for \( k = 1 : n - 1 \)
\( \beta = r_0 \oplus r(1:k) \odot y(1:k) \% 2 \)
\( \beta = \beta \oplus (\beta^*)^{-1} \odot \alpha^2 \% 1 \)
\( \mu = \beta^* \odot (r(k:-1:1) \odot x(1:k) \oplus b(k+1)) \)
\( v(1:k) = x(1:k) \oplus y(k:-1:1) \odot \mu \)
\( x(1:k) = v(1:k) \)
\( x(k+1) = \mu \)

if \( k < n - 1 \)
\( \alpha = \beta^* \odot (r(k:-1:1) \odot y(1:k) \oplus r(k+1)) \)
\( z(1:k) = y(1:k) \oplus y(k:-1:1) \odot \alpha \)
\( y(1:k) = z(1:k) \)
\( y(k+1) = \alpha \)
end
end

The computational complexity of all methods described in this section is \( O(n^2) \).

References


