A numerical search for intertwining relations

Jan M. Swart[*]

August 10, 2024

Abstract

We say that a probability kernel Q is intertwined on top of another probability kernel P if there exists a third probability kernel K such that $PK = KQ$. Given a probability kernel P that is a square matrix, we are interested in the problem of finding probability kernels Q and K of the same size as P such that Q is intertwined on top of P with intertwiner K. We describe a numerical method that is sometimes capable of finding such Q and K and present some numerical results.

Contents

[*]The Czech Academy of Sciences, Institute of Information Theory and Automation, Pod vodárenskou věží 4, 18200 Praha 8, Czech republic. swart@utia.cas.cz

1 Intertwining of probability kernels

We will be interested in relations of the form

$$PK = KQ, \tag{1}$$

where P, Q, and K are square matrices that are all of the same size $d \times d$ for some $d \geq 1$. A relation of the form (1) is called an *intertwining relation.* The matrix K is the *intertwiner.* If K is invertible, then we can rewrite (1) as

$$Q = K^{-1}PK \quad \text{or} \quad P = KQK^{-1}. \tag{2}$$

This relation says that P and Q are *similar.* It is known that this implies that P and Q have the same spectrum. By definition, the matrix P is diagonalisable if and only if K can be chosen such that Q is diagonal.

Recall that a $d \times d$ matrix P is a *probability kernel* if $P(x, y) \geq 0$ for all $1 \leq x, y \leq d$ and $\sum_y P(x, y) = 1$ for all $1 \leq x \leq d$. We will especially be interested in intertwining relations of the form (1) where P is a probability kernel. The n-th power of P then gives the n-step transition probabilities of the Markov chain that has P as its transition kernel. If we can diagonalise P, then we have very good control over the powers of P, since (2) implies

$$P^n = KQ^nK^{-1} \qquad (t \geq 0), \tag{3}$$

and it is trivial to calculate the powers of a diagonal matrix. A potential drawback is that typically, the matrices Q and K in (2) are not probability kernels, which means that diagonalisation ignores the very special property of P that it is a probability kernel. If we want to retain this information, then it may be advantageous to work instead with an intertwining relation of the form (1) such that not only P, but also Q and K are probability kernels. Then (1) implies that

$$P^nK = KQ^n \qquad (t \geq 0). \tag{4}$$

If we can find a relation between probability kernels of the form (1) such that Q is "simple" enough for us to have good control over its powers Q^n, then via (4) this gives information about the powers of the original probability kernel P that we are interested in. Thus, we can think of intertwining of probability kernels as an alternative to diagonalisation if we want to stay in the world of probability kernels.

As (2) shows, similarity of matrices is a symmetric relation: if P is similar to Q, then Q is similar to P. On the other hand, since the inverse of a probability kernel is usually not a probability kernel, in an intertwining relation between probability kernels of the form (1), P and Q do not play symmetric roles. To stress the difference, we will use the convention that if (1) holds, then we say that Q is intertwined "on top" of P, or that P is intertwined "below" Q. Given P, we will be interested in the problem of finding simple Q that are intertwined on top of P. One could similarly ask for Q that are interwined below P and much of what follows could be adapted to this case, but for concretness in this note we focus on Q that are intertwined on top of P.

2 The basic idea

We fix a square probability kernel P, i.e., an real matrix of size $d \times d$ such that

$$P(x,y) \geq 0 \quad \forall 1 \leq x,y \leq d \quad \text{and} \quad \sum_{y=1}^{d} P(x,y) = 1 \quad \forall 1 \leq x \leq d.$$

We consider the problem how to find a probability kernel Q of size $d \times d$ intertwined on top of P that is "as simple as possible". The basic idea is to define K_0 to be the identity matrix and then inductively define probability kernels $K_1, K_2, K_3, \ldots$ by

$$K_{t+1} := K_t + PK_t - K_t \mathcal{Q}(P, K_t) \qquad (t \geq 0), \tag{5}$$

for some cleverly chosen function $\mathcal{Q}$. If we are lucky, then there exist probability kernels K and Q such that

$$K_t \underset{t\to\infty}{\longrightarrow} K \quad \text{and} \quad \mathcal{Q}(P, K_t) =: Q_t \underset{t\to\infty}{\longrightarrow} Q, \tag{6}$$

which implies

$$PK - KQ = \lim_{t\to\infty} \big(PK_t - K_t Q_t\big) = \lim_{t\to\infty} \big(K_{t+1} - K_t\big) = 0, \tag{7}$$

leading to a solution of (1).

Recall that our aim is to find Q and K such that (1) holds. In this note we investigate the evolution (5) for a specific choice of the function $\mathcal{Q}$ that is based on the following two guiding principles:

- We want the kernel Q to be as simple in possible, in the sense that as many as possible of its off-diagonal elements are zero.
- We put restrictions on K by requiring some of its off-diagonal elements to be zero.

More precisely, we fix a set

$$Z \subset \big\{(x,y) \in \{1,\ldots,d\}^2 : x \neq y\big\} \tag{8}$$

that has the interpretation that these are the off-diagonal elements of K that we want to be zero, and define $\mathcal{Q} = \mathcal{Q}_Z$ as follows. First, we set

$$\begin{aligned}\mathcal{K}_Z := \big\{K : K &\text{ is a probability kernel of size } d \times d \\ &\text{such that } K(x,y) = 0 \text{ for all } (x,y) \in Z\big\},\end{aligned} \tag{9}$$

and we let $\mathcal{C}_Z(P,K)$ denote the set

$$\begin{aligned}\mathcal{C}_Z(P,K) := \big\{Q : Q &\text{ is a probability kernel of size } d \times d \\ &\text{such that } K' := K + PK - KQ \in \mathcal{K}_Z\big\}.\end{aligned} \tag{10}$$

For given probability kernels P, K of size $d \times d$ such that $K \in \mathcal{K}_Z$, we then define $\mathcal{Q}_Z(P,K)$ by setting

$$\mathcal{Q}_Z(P,K) := \text{the unique minimiser of } Q \mapsto \sum_{x \neq y} Q(x,y) \;\text{ in } \mathcal{C}_Z(P,K), \tag{11}$$

where the sum runs over all $1 \leq x, y \leq d$ such that $x \neq y$. The idea of minimising this function is that we want as many as possible of the off-diagonal elements of Q to be zero. It is should be noted that a priori, it is not clear that this is a good definition since in general we do not know whether such a minimiser exists (since $\mathcal{C}_Z(P, K)$ could be empty) or whether the minimiser is unique. We will not really solve this problem but just hope that in the numerical examples that we will investigate things will turn out to be OK.

It should be noted that there is one case where it is clear that the evolution in (5) is well-defined. This is the case when $Z = \emptyset$, i.e., the case when we do not require any of the off-diagonal elements of K to be zero. In this case the function in (11) has a unique minimiser, which is the identity matrix. As a result, in this case, the evolution equation (5) reduces to

$$K_{t+1} = PK_t \qquad (t \geq 0), \tag{12}$$

which together with the initial condition $K_0 = 1$ means that $K_t = P^t$ $(t \geq 0)$. This does in general not lead to interesting intertwining relations, which is the main reason why we require some of the off-diagonal elements of K to be zero.

3 Numerical implementation

3.1 Basic usage of the scripts

A couple of scripts, written in the scientific programming language GNU Octave, allow one to numerically solve the evolution equation (5) for a given probability kernel P and choice of the set Z from (8). These scripts are available from my homepage [Swa24]. The basic scripts are `step`, which calculates one step of the evolution (5), and `evolve`, which runs the evolution (5) until an intertwining is found within the required precision. Before one can run these scripts, the following variables must be defined:

- `P`. A square matrix of size $d \times d$ for some $d \geq 1$ that is the probability kernel that we want to intertwine.

- `Z`. A square matrix of size $d \times d$ containing only zeros and ones, where the ones represent the set Z from (8).

Instead of setting `P` and `Z` by hand, one can also load any of a number of datasets with names of the form `setup_xxx.dat` that contain predefined variables `P` and `Z` of a certain form. Alternatively, one can run any of a number of scripts with names of the form `setup_xxx.m`. Some of these scripts have optional parameters such as the size d of P. This is explained in the comments at the beginning of these scripts.

The script `evolve` numerically solves the evolution equation (5) and calculates in each step the quantity

$$\varepsilon_t := \sup_{1 \leq x, y \neq n} \big| PK_{t+1}(x, y) - K_{t+1}Q_t(x, y) \big| \qquad (t \geq 0). \tag{13}$$

The script `evolve` runs until one of the following conditions is satisfied:

- $\varepsilon_t \leq$ `tol`,

- the program has been running longer than `hour` hours, `minu` minutes, and `seco` seconds.

The variables `tol`, `hour`, `minu`, and `seco` are optional. Their default values are

$$\texttt{tol} = 10^{-7}, \quad \texttt{hour} = 0, \quad \texttt{minu} = 0, \quad \texttt{seco} = 3.$$

Before running `evolve`, they can be set to different values. If `hour` or `minu` are set to a nonzero value, then the default value of `seco` is zero.

The script `evolve` produces the following output:

- `Q` A probability kernel of size $d \times d$ that corresponds to Q_t where s is the last step of the evolution.
- `K` A probability kernel of size $d \times d$ that corresponds to K_{t+1} where s is the last step of the evolution.
- `erro` The quantity ε_t defined in (13) where s is the last step of the evolution. If this is small, then the program has found an approximate intertwining relation.

3.2 The way the scripts work

Let P and K be probability kernels of size $d \times d$, let Z be a set as in (8), and assume that $K(x, y) = 0$ for all $(x, y) \in Z$. Then we claim that the space $\mathcal{C}_Z(P, K)$ defined in (10) is the set of all $d \times d$ matrices that satisfy

$$\begin{array}{lll} \text{(i)} & Q(x,y) \geq 0 & (1 \leq x, y \leq d), \\ \text{(ii)} & \displaystyle\sum_{y=1}^{d} Q(x,y) = 1 & (1 \leq x \leq d), \\ \text{(iii)} & KQ(x,y) = K(x,y) + PK(x,y) & \big((x,y) \in Z\big), \\ \text{(iv)} & KQ(x,y) \leq K(x,y) + PK(x,y) & \big((x,y) \in [d]^2 \backslash Z\big). \end{array} \tag{14}$$

Indeed, (i) and (ii) say that Q is a probability kernel while (iii) and (iv) guarantee that $K' := K + PK - KQ$ is a nonnegative matrix that satisfies $K'(x, y) = 0$ for all $(x, y) \in Z$. The latter then implies that K is a probability kernel since the relation $\sum_y K'(x, y) = 1$ $(1 \leq x \leq d)$ is an automatic consequence of the fact that P, K and Q are probability kernels. As a result, we see that given Z, P, and K, to calculate $Q = \mathcal{Q}_Z(P, K)$ as defined in (11), one needs to solve the following problem:

$$\text{Find the minimiser of } Q \mapsto \textstyle\sum_{x \neq y} Q(x,y) \text{ subject to the constraints (14)} \tag{15}$$

Note that at most d^2 of the constraints (14) are equalities while the space of $d \times d$ matrices has dimension d^2, so it is reasonable to hope that there is at least one minimiser.

The linear optimisation problem (15) is a standard exercise in linear programming for which we use the predefined Octave function `glpk`. The script `evolve` calls the function `Qfunc`, which in turn calls the function `Lipo`, which is based on the predefined function `glpk`. The description of the functions `Lipo` and `Qfunc` is as follows:

> `Lipo(A,b,c,I)`. The input are: a matrix A of size $d \times d$, a column vector b of length N, a row vector c of length d, and a row vector I of length N containing only zeros and ones. The output is a column vector x of length d that minimises the function $x \mapsto c \cdot x$ subject to the constraints $x(k) \geq 0$ for all k, $Ax(k) = b(k)$ for all k with $I(k) = 1$, and $Ax(k) \leq b(k)$ for all k with $I(k) = 0$.

Qfunc(P,K,Z). The input are: probability kernels P and K of size $d \times d$ and a matrix Z of size $d \times d$ containing only zeros and ones such that $K(x, y) = 0$ for all (x, y) with $Z(x, y) = 0$. The output is a solution Q of the linear optimisation problem (15).

As described in Subsection 3.1, the script `evolve` iterates the evolution equation (5) until it has found an approximate intertwining relation within the set tolerance or time is up. To counter the effect of small numerical errors that tend to multiply, in each step of the evolution, small corrections to K are carried out to ensure that it remains a probability kernel.

4 Results

4.1 An intertwining of birth-and-death chains

Loading `setup_birth1.dat` or running `setup_birth1` creates matrices `P` and `Z` of the following form:

$$\mathtt{P} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathtt{Z} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{16}$$

Running `evolve` now produces the matrices

$$\mathtt{Q} = \begin{pmatrix} 0.9619 & 0.0381 & 0 & 0 & 0 \\ 0 & 0.6913 & 0.3087 & 0 & 0 \\ 0 & 0 & 0.3087 & 0.6913 & 0 \\ 0 & 0 & 0 & 0.0381 & 0.9619 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{17}$$

and

$$\mathtt{K} = \begin{pmatrix} 1.0000 & 0 & 0 & 0 & 0 \\ 0.9239 & 0.0761 & 0 & 0 & 0 \\ 0.7071 & 0.1989 & 0.0940 & 0 & 0 \\ 0.3827 & 0.1838 & 0.1737 & 0.2599 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 \end{pmatrix} \tag{18}$$

and the calculated error is $\mathtt{erro} = 9.8 \cdot 10^{-08}$. As one can verify by typing `sort(eig(P))`, the values on the diagonal of `Q` are the eigenvalues of `P`. Since

$$P^n K = K Q^n \qquad (t \geq 0), \tag{19}$$

the following two procedures have the same effect:

1. Starting from an initial state in $\{1, \ldots, 5\}$, first evolve the state for n time steps according to the Markov chain with transition kernel P, then map the final state into a new state according to the probability kernel K.

2. Starting from an initial state in $\{1, \ldots, 5\}$, first map the initial state into a new state according to the probability kernel K, then evolve the state for n time steps according to the Markov chain with transition kernel Q.

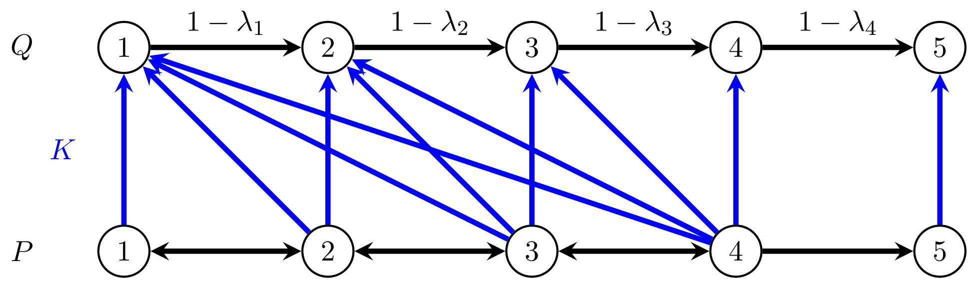

Figure 1: Interwining for a birth-and-death process with a trap. The eigenvalues of P are $1 = \lambda_0 > \lambda_1 > \cdots > \lambda_4$.

We have symbolically depicted this intertwining relation in Figure 1, where black arrows indicate positive transition probabilities of the kernels P and Q and blue arrows indicate positive transition probabilities of the kernel K. (For better readability of the pictures, we have not drawn jumps from a point to itself.) If before running `setup_birth1`, one sets the value of the parameter `n` to a different value than 5, then one can verify that analogue intertwining relations hold for transition kernels P of birth-and-death Markov chains with state spaces of a different size. The script `setup_birth1r.m` produces random `P` of the form

$$\mathtt{P} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2}r_2 & \frac{1}{2} & \frac{1}{2}(1-r_2) & 0 & 0 \\ 0 & \frac{1}{2}r_3 & \frac{1}{2} & \frac{1}{2}(1-r_3) & 0 \\ 0 & 0 & \frac{1}{2}r_4 & \frac{1}{2} & \frac{1}{2}(1-r_4) \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{20}$$

where $r_2, \ldots, r_4$ are independent uniformly distributed on $[0,1]$. Also in this case, `evolve` finds an intertwining relation of the type depicted in Figure 1.

In [Swa10], intertwining relations of the type depicted in Figure 1 have been theoretically derived in the case that $P = P_t$ is the transition kernel of a continuous-time birth-and-death process. This built on earlier work in [DM09] which was concerned with Q that are intertwined *below* P, rather than on top of it. Based on the numerical evidence, we conjecture that such intertwining relations exist quite generally for probability kernels P on sets of the form $\{1, \ldots, d\}$ that satisfy:

(i) $P(x,y) = 0$ for all $|x-y| \geq 2$,

(ii) $P(n,n) = 1$,

(iii) the spectrum of P is contained in $[0,1]$.

Condition (iii) is satisfied for lazy kernels, i.e., P of the form $\frac{1}{2}(P' + 1)$ where P' is another probability kernel. Condition (iii) is necessary for intertwinings of the type depicted in Figure 1 since for such intertwinings the eigenvalues $\lambda_1, \ldots, \lambda_4$ need to be probabilities. Condition (iii) may fail in general for birth-and-death kernels. A example is:

$$\mathtt{P} = \begin{pmatrix} \frac{1}{2} & \frac{2}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{21}$$

which has an eigenvalue $\lambda_4 \approx -0.282586$.

4.2 A different intertwining of birth-and-death chains

Loading `setup_birth2a.dat` or running `setup_birth2` creates matrices `P` and `Z` of the following form:

$$\mathtt{P} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathtt{Z} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \tag{22}$$

Note that this differs from (16) only in the choice of Z. Running `evolve` now produces the matrices

$$\mathtt{Q} = \begin{pmatrix} 0.7706 & 0.2294 & 0 & 0 & 0 \\ 0.2257 & 0.5560 & 0.2183 & 0 & 0 \\ 0 & 0.2025 & 0.6353 & 0.1622 & 0 \\ 0 & 0 & 0 & 0.0381 & 0.9619 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{23}$$

and

$$\mathtt{K} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.5412 & 0.4588 & 0 & 0 & 0 \\ 0 & 0.5995 & 0.4005 & 0 & 0 \\ 0 & 0 & 0.7401 & 0.2599 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{24}$$

and the calculated error is $\mathtt{erro} = 7.96 \cdot 10^{-08}$. This intertwining is depicted in the lower part of Figure 2. As indicated in that figure, one can find probability kernels Q_0, Q_1, Q_2, Q_3 and K_1, K_2, K_3 such that $Q_0 = P$ and $Q_{i-1}K_i = K_iQ_i$, which implies that

$$PK = Q_0K_1K_2K_3 = K_1Q_1K_2K_3 = K_1K_2Q_2K_3 = K_1K_2K_3Q_3 = KQ, \tag{25}$$

where $K := K_1K_2K_3$ is the intertwiner from Subsection 4.1. By setting d to a different value than the default $d = 5$ before running `setup_birth2` one can check that similar intertwinings hold for birth-and-death kernels of a different size, and with the script `setup_birth2r` one can investigate randomly created birth-and-death kernels. Intertwining relations of the form depicted in Figure 2 seem to be new, as they do not occur in [Swa10].

It is not completely clear in what generality intertwining relations of this form exist. The script `setup_birth2r` sometimes produces probability kernels P for which `evolve` finds an

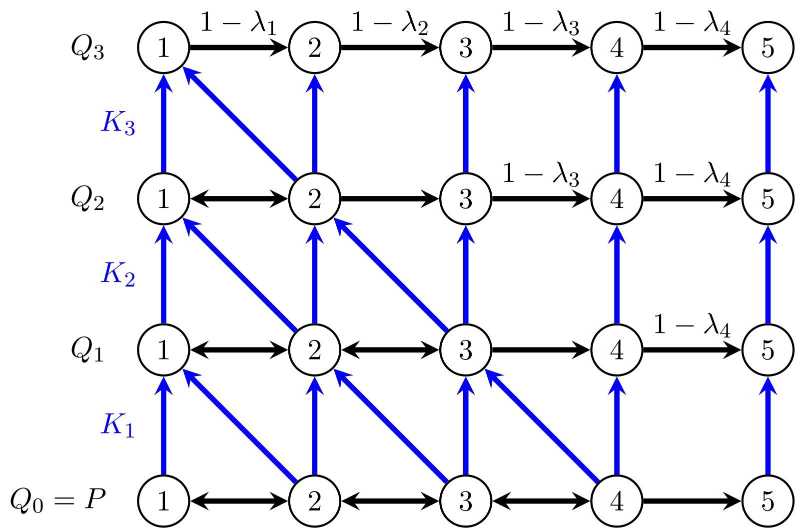

Figure 2: The intertwining from Subsection 4.1 can be obtained as a concatenation of intertwinings of the form described in Subsection 4.2. The kernels Q and K from Figure 1 are given in terms of those in the present figure by $Q = Q_3$ and $K = K_1K_2K_3$.

intertwining relation with a probability kernel Q of a different form. To see an example of this, load `setup_birth2b.dat` and then run `evolve`.

5 Open problems

The general problem of finding intertwining relations for a given probability kernel, as a probabilistic alternative to diagonalisation, seems interesting. Before trying to attack this problem theoretically, it is probably wise to investigate it numerically first, to get a rough idea of the kind of statements that one might want to prove. We have demonstrated that the evolution equation (5) together with the choice of the function $\mathcal{Q} = \mathcal{Q}_Z$ in (11) can sometimes yield interesting numerical results for transition kernels P birth-and-death chains. For lack of time, we have not investigated other P, but it would seem interesting to do so.

A disadvantage of the definition of the function $\mathcal{Q} = \mathcal{Q}_Z$ in (11) is that there is no guarantee that the minimiser is exists or is unique. To see that the minimiser can sometimes fail to exist, run `setup_contact` and then run `step` two times. In this example, that is inspired by a contact process on a lattice of size two, in the first step, Q_1 can still be calculated, but in the second step the set $\mathcal{C}_Z(P, K_1)$ (numerically) turns out to be empty which means that the evolution (5) is not defined from this point on.

Clearly, there are a lot of alternative approaches one can try. One can try to tweak the definition of $\mathcal{Q} = \mathcal{Q}_Z$ in (11) or try to replace the evolution equation (5) by something different altogether. For example, an obvious thing one could try is to find probability kernels K and Q that minimise the quadratic function

$$(K, Q) \mapsto (PK - KQ)^\dagger (PK - KQ), \tag{26}$$

where $A^\dagger$ denotes the transpose of a matrix A, with the idea that the function in (26) assumes its minimal value zero exactly on those pairs (K, Q) for which the intertwining relation (1) holds. By requiring certain elements of K and Q to be zero one can then try to restrict the space of solutions and search for intertwining relations of a certain form.

References

[DM09] P. Diaconis and L. Miclo. On times to quasi-stationarity for birth and death processes. *J. Theor. Probab.* 22 (2009), 558–586.

[Swa10] J.M. Swart. Intertwining of birth-and-death processes. *Kybernetika* 47(1) (2011), 1–14.

[Swa24] J.M. Swart. A numerical search for intertwining relations. Research note and collection of scripts. Available from: `https://staff.utia.cas.cz/swart/res_index.html`.